Part I

Data Modeling and Camera Characterization

Abstract

In this part, the procedure we used to model the noise characteristics of a digital CCD camera is described in detail. The functioning of a CCD is described, along with the various sources of noise present in the camera system. A systematic procedure is developed to measure the spatial and temporal noise of the camera, and the results are shown in detail. Finally, the measurement of spatial frequency response of the camera system and the validation of various noise models are proposed as future work.

Chapter 1

Introduction and Motivation

This part of the report describes the progress of research at Penn State's Computer Vision Lab to develop a simple but accurate method for characterizing and removing the noise introduced by a digital CCD camera. Digital CCD cameras offer superior performance as compared to their analog counterparts. For example, digital cameras are free of the spatial inconsistencies between rows and between frames (i.e., jitter) that may be caused by video clock instabilities. By its nature, a digital imaging system is also highly immune to the spatial and temporal artifacts that may be introduced by transmission-line noise. As noted in Section 3 below, however, several noise processes may still be encountered in such a system. The goal of the modeling and characterization of the camera described here is to enhance the operation of a system for airborne obstacle detection. As an example, consider a Cessna aircraft that has a length and wing-span of approximately 9 m (30') and the fuselage diameter of approximately 1.2 m (4') [5]. The detection algorithm must be capable of detecting this small target at least 25 seconds prior to a possible collision to allow for corrective actions by the pilot. Assuming that both aircrafts are traveling at 125 m/s (250 knots), their relative velocity can be as high as 250 m/s (500 knots). In such case, they would be 6.25 km (3.5 nautical miles) apart 25 seconds before collision. Using a camera with a resolution of 60 pixels per degree, the image of the aircraft is of 5.0 x 0.7 pixels from a side view, but only 0.7 x 0.7 pixels from a front view. It is clear that safety demands that noise effects even at the sub-pixel level must be accounted for and compensated as much as possible.

Chapter 2

CCD Operation and Noise Sources

This chapter describes the theory of operation of a CCD camera, followed by the description of various noise sources affecting a CCD camera.

2.1 Basics of CCD theory of operation

In this section, the theory of CCD operation is presented, and terms are defined. Variants of CCD architecture are compared, and refinements such as blooming suppression are explained [2]. Discussion will lead to specific features of Kodak ES 1.0 camera.

2.1.1 Theory of CCD operation

The charge-coupled device (CCD) first appeared in a 1970 Bell Labs technical report. Its usefulness in both analog and digital electronics was recognized at once, and CCDs have been used, for example, in signal processing applications such as delay lines for both analog and digital signals. Since about 1980, however, the term CCD has become synonymous with video imaging for both the mass-produced consumer and the top-performance scientific markets. Although CMOS imaging devices, offering a one-chip solution to image capture and processing, are about to enter the consumer market, it seems certain that CCDs will continue to dominate the high-performance imaging market for some time to come.

To start with, it should be noted that the CCD is an analog device, and not a digital one. It is true that the operative quantities in the CCD are charges, and that because these charges occur in quantizable form as electrons, there is a discrete character to the device's operation. Any semiconductive device, however, operates by the transfer of charge carriers.

Although at the lowest level all such operation is discretizable, it is only when we associate an information content to a transition between discrete levels of much greater magnitude – a transition that is largely unaffected by the noise processes inherent in all such physical systems – that we call such a device 'digital.'

In its simplest form, a CCD comprises an array of charge storage sites, wherein each storage site is an MOS capacitor as shown in Figure 2.1 (a). An MOS (for metal-oxide-semiconductor) capacitor comprises an insulating layer of silicon oxide sandwiched between a metallic (e.g. aluminum) gate and a silicon substrate, which has been doped into semiconductivity with an excess of p-type carriers (holes). Typically, the gate is made of degeneratively doped polycrystalline silicon (or polysilicon) instead of a metal. Output leads are bonded via ohmic contacts to the gate and the substrate.

When a potential is applied between the gate and substrate, a region develops in the substrate underlying the gate that is swept free of p-type carriers by electrostatic repulsion (Figure 2.1 (b)). Any electrons that may appear in this region (e.g., via injection or generation) will be attracted to the gate and thus will congregate below the oxide layer. Because the p-type carriers have been repelled from this region, the electrons are protected against recombination, and the quantity of charge which they represent in the aggregate – called a charge packet – will be preserved indefinitely. This region is called the depletion region, and the electrostatic barrier that defines it is called a potential well.

Each storage site can hold only a finite number of electrons before it begins to overflow. This number is called the 'full-well' capacity and generally (for CCD imagers) ranges from about 20,000 to about 100,000.

Once the charge packet has been formed, it remains to pass the packet along the array from one storage site to the next without altering its contents. The information represented by the value of the charge packet cannot be known until that value can be outputted from the chip. The basic and most common mechanism for the transfer of charge packets is the three-phase clocked approach shown in Figure 2.2. The first phase is the application of a potential to the A sites, creating potential wells. The second phase is to apply a potential to the B sites as well, thereby spreading the charge packet between the A and B sites. The third phase is to remove the potential from the A sites, which completes the process of moving the charge packets from the A sites to the B sites. In the next three phases, the packets are moved from the B sites to the C sites, and so on. From this description, one may understand why early CCD delay lines were referred to as 'bucket brigade devices.'

Although one of every three sites is not used during each cycle, one may see that this

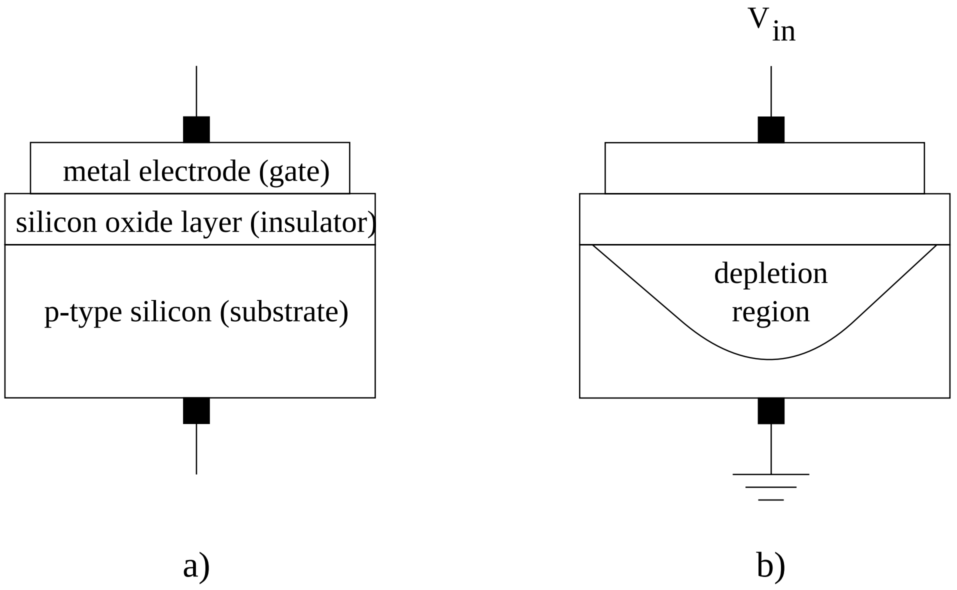


Figure 2.1: (a) A Metal Oxide Semiconductor (MOS) capacitor. (b) Depletion region in the MOS capacitor, when a potential is applied between the gate and the substrate.

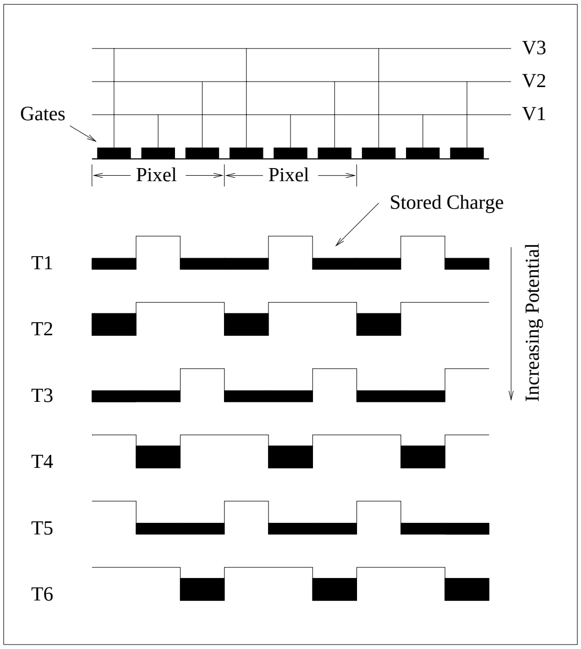


Figure 2.2: Charge transfer in a three-phase device. This represents one column. Rows go into the paper. Six steps are required to move the charge one pixel [2].

empty site provides the necessary function of separating each charge packet from the next one. Likewise, one may see that when a three-phase clocking method is used, the available site size (and therefore the maximum site capacity) is reduced to one-third. For video applications, for example, each pixel must be broken into three areas, only one of which may be optically active, so sensitivity is necessarily reduced.

Because of the capacity and sensitivity constraints of the three-phase system, other approaches have also been developed. By changing only the clocking method, for example, a four-phase approach may be used which allows an adjacent two of every four sites to be active at a time, thereby increasing pixel capacity to one-half. One may obtain two-phase operation by modulating the thickness of the oxide layer, and chip designs that permit one-phase clocking also exist. It is believed that our Kodak MEGAPLUS ES 1.0 camera uses a three-phase clock, but the engineers we spoke to could not definitively confirm this feature.

In a two-dimensional array, the packets in each column are transferred as described above, each column acting independently but in synchronization with all of the others. Each column empties into a shift-register row which operates in the same fashion as the columns but on a different timing scheme. That is, every time the column charge packets travel one site down the array, the column transfer operation must pause while the entire shift register is transferred sequentially through the terminal site. This terminal site, the last site on the array, is a diode which converts each incoming charge packet into a potential (i.e., a voltage). The stream of varying potentials may be amplified before being outputted from the chip as a raster image signal for further processing (including digitization).

2.1.2 CCD imaging architectures

In an imaging CCD, the charge packets are created by the photoexcitation of bound electrons into a free state by incident photons, and the subsequent migration of these free electrons into the depletion region. (As an aside, we note that the depletion region is usually so shallow that few free electrons are actually generated within it.) So long as light is incident on the array, this process will continue. One may easily realize that the continuation of this process after image capture and during the transfer of the charge packets would cause image degradation.

One solution to this problem is the use of a mechanical shutter synchronized to the capture/transfer timing. A better solution, called 'electronic shuttering,' uses different areas of the chip for capture and transfer. The transfer gates are covered with an opaque mask

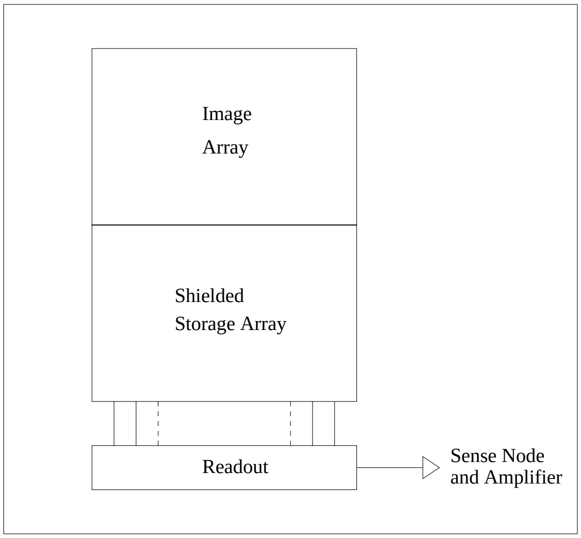


Figure 2.3: Basic architecture for a full frame CCD [2].

so that the packets being transferred will not be corrupted. As a consequence, the array may be illuminated constantly without affecting the transfer process, thus increasing camera sensitivity. On the other hand, it should be noted that division of the chip into two areas necessarily decreases the area used for image capture, thereby decreasing camera sensitivity.

At least two main divisional configurations exist. In a frame transfer configuration, the active and shielded areas of the chip are completely separated, as shown in Figure 2.3. At the end of the capture period, the packets from all of the active sites are simultaneously dumped into corresponding sites in the shielded array, and the transfer process begins. Note that a new capture period may begin at the same time the transfer process begins. Because of the physical concentration of the active sites and the high area sensitivity that results, high-performance cameras for scientific applications usually contain frame transfer CCDs.

In the interline transfer configuration, lines of storage sites for image transfer are fabricated next to each line of active sites, as shown in Figure 2.4. Photodiodes rather than MOS capacitors are most often used for the active sites in such arrays (as is the case in our Kodak camera). The main disadvantage of this configuration is that as little as 20 % of the chip area may be available for image generation, resulting in a severe loss of sensitivity. For this reason, a microlens array is usually positioned adjacent to the chip surface to increase the 'fill factor.' The microlens array contains one lens for each pixel, which focuses the light incident on the entire pixel onto the area of the active site.

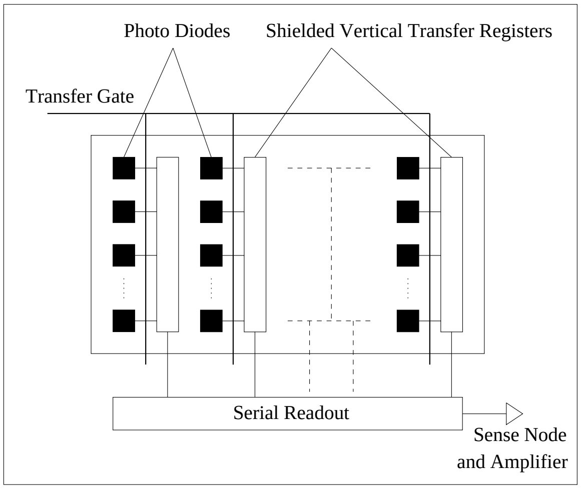


Figure 2.4: Interline transfer architecture. The charge is rapidly transferred to interline transfer registers via the transfer gate. The registers may have three or four gates [2].

2.1.3 Other features of CCD camera chips

If the depletion region of an active site is filled beyond capacity, the excess charge will spill over and contaminate adjacent sites in a process called 'blooming.' Spill-over between adjacent columns is prevented during fabrication, but a particularly strong local illumination may saturate most of the length of the affected columns. Blooming suppression combats this effect with anti-blooming drains that suddenly flatten the response of the active site above a certain intensity.

In some cameras, such as our Kodak camera, the bias point of the anti-blooming drains is variable. While reducing the bias point necessarily reduces the array sensitivity, in some applications such a tradeoff may be acceptable. In cameras used to monitor automobile traffic, for example, increased resistance to blooming caused by headlights may be worth some loss of sensitivity.

Our Kodak camera also has a dual-channel transfer configuration. In this structure, the even and odd rows of the array are processed and outputted through two separate channels. This configuration allows faster data throughput: our camera can supply 30 frames of 1000×1000 pixels per second, while a single-channel version of the same camera can only supply 15 frames per second. With respect to noise analysis, however, it is important to recognize the fact that pixels in adjacent rows may not be subject to the same noise processes at any given time. This disparity is especially important when considering frequency-dependent processes, such as the Modulation Transfer Function (MTF) as discussed in Section 8 below.

2.2 Noise sources in a digital CCD camera

In this section, various types of noise sources in a CCD camera are discussed [2]. Dark and photoresponse noises are distinguished. CCD-generated noise (e.g. fixed pattern noise) and noise generated by support electronics (e.g. readout noise) are distinguished. Other noises such as interpixel effects (e.g. blooming and smear) and optical effects such as point-spread functions are also discussed.

2.2.1 Overview of relevant noise processes

Although in-camera digitization offers good protection against transmission-line noise, the signal outputted by the camera is only a flawed representation of the image which is incident on the CCD array. For one thing, the photosites are like snowflakes in that none is exactly like any one of the others, and each site will respond somewhat differently to the same level of luminous excitation.

Likewise, in any such device that is operating above absolute zero, electrons are generated thermally as well as optically. Once generated, each electron is indistinguishable from any other, so some portion of the charge packet is necessarily always invalid. Moreover, each site responds differently to this noise process as well.

Finally, the on-chip paths by which the charge packets are read out from the chip and converted into potential values, and the off-chip circuitry through which these signals are amplified and digitized, introduce errors of their own that may vary with signal amplitude and frequency. An illustration of the imaging system path and some of the noise processes associated with each step is presented in Figure 2.5.

2.2.2 Dark-field response and nonuniformities

As mentioned above, the electrons that migrate into the depletion region may be generated by thermal as well as photoelectric processes. Therefore, some signal will be outputted even when the array is in total darkness. The result of this phenomenon is called the camera's dark-field response.

Dark-field response will vary from pixel to pixel. This noise process is also extremely temperature-dependent: for example, the noise level doubles when the array temperature

Atmosphere	Lens	Camera			
		CCD		Support Electronics	
Point-spread function (PSF) (a linear function of signal frequency	PSF cos^4 effect	1) Localized effects: Dark noise, Photoresponse nonuniformities 2) Interpixel effects: Blooming, Charge transfer efficiency	Reset (kTC) noise	Readout noise (e.g. from amplifiers	Quantization noise

Figure 2.5: An illustration of the imaging system path and some of the noise processes associated with each step

increases by 8 to 9 degrees Celcius. For this reason, measurements should be taken only after the camera has warmed up. (In order to obtain an accurate measurement of the dark field, we believe that it is also important to allow the array to warm up under normal operating illumination conditions. A focused image of any intensity will certainly affect the surface temperature of the array, and thereby influence the dark component of the total response.)

The magnitude of the dark-field response is also, of course, linearly dependent on the exposure time, i.e., the period of time during which electrons are being collected. Although thermal electrons may also potentially corrupt the charge packets during the transfer process, transfer across the chip occurs so rapidly that this quantity is usually ignored.

Fortunately, the thermal noise process is simply additive. So long as we can reliably estimate the number of such electrons collected at a particular site, it is a trivial matter to subtract that measure from the gross response.

CCD chips are usually (if not universally) fabricated so that some of the active sites on the periphery are shielded from illumination. (An area of isolation pixels also separates these dark pixels from the active ones in order to prevent light leakage.) During image processing, the values returned by the dark pixels may be used to calculate an estimate of the magnitude of the array's dark response, which may then be subtracted from the outputted image. As the response of each active site is unique, however, the accuracy of this approach to dark-field compensation is not optimal.

Our Kodak camera includes a feature called ‘dark-clamping’ whereby such an estimate of the dark field is automatically subtracted from the image. While not exactly accurate and therefore not entirely appropriate for our present purposes, this feature is considered a significant advantage for consumer applications. In ‘dark-clamping,’ the dark-field estimate is automatically calculated and subtracted, so that the camera’s output has already been compensated. As the process is transparent and occurs directly at the chip output, it is not necessary to keep track of exposure time or temperature. It may be possible to disable this function in our camera via a software command, but the particular Kodak engineers with such knowledge have so far been unresponsive to our requests.

2.2.3 Photoresponse nonuniformity

Just as the active sites vary in their response to thermal excitation, they also vary in their photoresponse. In other words, each pixel will react differently to the same level of incident illumination. The degree by which the number of photoelectrons collected by a particular site varies from an arbitrary standard amount may be thought of as a local ‘gain factor,’ as this noise process is multiplicative with respect to the level of excitation and the response of each site is generally quite linear.

The dark-field and photoresponse nonuniformities together comprise the array’s ‘fixed-pattern noise.’ Generally, fixed-pattern noise is defined only at each pixel and has no spatially varying component. In other words, there will be no correlation between the fixed-pattern noise at two adjacent pixels. However, we note that there will usually (if not always) be a low-frequency component to the photoresponse nonuniformity, caused by unavoidable variations in the substrate thickness. These variations cause photons of the same wavelength to interact differently at the quantum level at different points on the array. This effect is not a separate factor, though, and is incorporated into the general photoresponse nonuniformity.

2.2.4 Charge-transfer efficiency

Although each transfer gate successfully moves well over 99.99 % of each charge packet to the next gate in the column, some small amount of charge stays behind. When the incident image contains sharp (i.e., high-frequency) transitions between areas varying greatly in amplitude, this process will tend to filter out the high spatial frequencies by blurring the transitions. This effect increases with array size, i.e., with the number of transfers required to move each packet off of the array. Charge-transfer efficiency is a component of the array’s frequency response and therefore is included in the array's modulation transfer function (Section 8).

2.2.5 Other interpixel noise mechanisms

As discussed in Section 2 above, blooming occurs when the charge in a saturated pixel overflows into adjacent pixels in the column. By the anti-blooming measures described above, the post-saturation response can be largely reduced, but it cannot be eliminated. The extent to which a charge packet has been corrupted by overflow is of course indeterminable, and in processing the resulting images this effect must be kept in mind. Whenever a saturated pixel is encountered, the signal outputted from its column neighbors must be considered suspect and possibly corrupted.

Another source of interpixel noise, called 'smear' (or sometimes tunneling), occurs when photoelectrons generated at one site migrate instead into a neighboring site. (This process is quite different from the process, also called 'smear,' which occurs when transfer occurs in a non-shielded array while the array is still being illuminated.) As this process is related to signal frequency, we would expect it to be included in the array's modulation transfer function (Section 8).

2.2.6 Reset noise

Reset noise arises when the capacitor which converts the charge packets into potential values is not completely reset between packets. As this noise is highly temperature-dependent, it is also referred to as kTC noise, where k is the Boltzmann constant, T is the temperature and C is the capacitance of the device. In most if not all CCDs manufactured today, this noise process has been virtually eliminated through the use of correlated double sampling (CDS), whereby two samples are taken from each packet and averaged to remove the reset error.

2.2.7 Readout noise

As the signal generated by the CCD array is exceedingly small, it must be amplified before processing. Each of the amplification and processing stages necessarily introduces its own noise process, which will generally be dependent on temperature and signal frequency. On the whole, though, this noise is random in time and cannot be compensated.

2.2.8 Quantization noise

Conversion of the analog array signal into a digital quantity necessarily results in a certain loss of information. While this noise is completely random and unknowable, it is easily characterized as a zero-mean process whose variance is a function of the number of bits in the digital output.

Chapter 3

Estimation of CCD Noise

This chapter describes the methods for estimating the parameters of the temporal as well as the fixed pattern noise of a CCD camera. Healy-Kondepudy noise estimation procedure is used to estimate the temporal noise. An experimental protocol is developed for estimation of fixed pattern noise, and the detailed mathematical analysis for least squares estimation of the noise is presented. Results of noise estimation using the Kodak ES 1.0 camera are described.

3.1 Description of Healey-Kondepudy noise estimation procedure

Following a common model of CCD behavior, Healey and Kondepudy [1] express the digital output D at each pixel as

$$D = (KI + E_{DC} + N_S + N_R)A + N_Q,$$

where

- K is a factor that characterizes the pixel's photoresponse,
- I is the incident illumination,
- E_{DC} is the expected number of dark electrons,
- N_S is the shot noise,
- N_R is the readout noise,

- A is the analog gain, and
- N_Q is the quantization error.

To reduce this expression, Healey and Kondepudy make the following three assumptions:

1. The photoresponse factor K is very close to 1 for all pixels,
2. The expected number of dark electrons E_{DC} is nearly constant across the array, and
3. The incident illumination I is nearly constant across the array.

Using these assumptions, and representing the image-wide means of I and E_{DC} as $\bar{I}$ and $\bar{E}_{DC}$, the expression for D is reduced to the form

$$D = \mu + N,$$

where

$$\mu = A(\bar{I} + \bar{E}_{DC})$$

and N is a zero-mean random variable characterized by

$$\sigma_N^2 = A^2(\bar{I} + \bar{E}_{DC}) + \sigma_C^2.$$

Here the noise term σ_C^2 is assumed independent of the number of collected electrons:

$$\sigma_C^2 = A^2\sigma_R^2 + \frac{q^2}{12}.$$

These relations imply

$$\sigma_N^2 = A\mu + \sigma_C^2,$$

so by taking the differences between pairs of similar images (i.e. $\mu_1 = \mu_2$), Healey and Kondepudy estimate the parameters A and σ_C^2.

3.2 Development of pattern noise (flat-field) experimental protocol

In this section, a history of what we have observed is presented, including the drawbacks of our previous setups, and concluding with a detailed description of the final test bench. Possible sources of error are noted, for example, the failure to consider spectral content, and the use of neutral density filters in the optical path.

3.2.1 Fluorescent sources

As a first step, we resolved to determine the flat-field response of our camera: i.e., the level of interpixel variation when each site in the entire array was presented with the same level of excitation. Because we had a relatively large computing capacity available, it seemed that we could capture and process large numbers of images fairly quickly, averaging the responses over time in order to eliminate temporal variations, and thereby develop a flat-field model that could easily be verified. Obtaining a field illumination that was uniform in both time and space, however, turned out to be problematic, as those with more optical engineering experience might already know.

First, we concentrated on using reflected excitation. We reasoned that if transmitted light were used, it would be impossible to completely remove the image of the light source from our field, even through diffusing sheets and a defocused lens. Therefore, reflection from a Lambertian surface appeared more promising in this regard.

Although indoor fluorescent lighting is prevalent and apparently very uniform, it immediately became obvious that a fluorescent lighting source would not be suitable. The accuracy of our results, and specifically their immunity to temporal variations, would depend on our ability to collect a large number of images under identical excitation conditions. Conventional fluorescent lights, of course, flicker at approximately a 60 Hz rate, rendering the level of illumination across any sequence of shuttered images very non-uniform and therefore unusable for our purposes.

High-frequency fluorescent sources are available, being priced at about $1,000 for a 10-inch square diffuse source operating at 15 kHz or higher. Even through a diffusing layer, however, such sources are not uniform enough to present a transmitted flat-field, and would have to be used as target illuminators. We soon discovered that the problem of evenly illuminating a diffusing target was not trivial, so this approach was not an optimal solution either. (For a target, we used an opaque sheet of coated matte paper that was supplied as a protective spacer for laser-printer color transparency blanks.)

We did obtain good uniformity using the blank screen of a laptop computer as a transmitted flat-field. The internal configuration of such a device is unknown to us, but the operating frequency seemed to be high enough to provide temporal uniformity. It was impossible to vary the brightness of this field while maintaining the spatial uniformity, though. Although when white the pixels were uniform, their brightness varied from row to row when they were darkened. Also, the overall intensity of the source was insufficient to permit the use of neutral density filters to obtain different brightness levels. Although such a source is convenient and generally available, the difficulty of evenly varying its brightness makes it unsuitable for the purposes of our model.

3.2.2 Incandescent sources

We found that an AC-powered incandescent source was also not entirely free from temporal flicker. While DC-powered incandescent sources are available, we also found it impossible to obtain spatially uniform illumination from such a source. We tried bulbs mounted in reflectors and an overhead projector, each shone through a diffusing sheet, but were unable to completely remove the filament image from the illumination.

Consultation with a Kodak research scientist gave us some insights into general optical laboratory practices. We learned that DC-powered incandescent sources may be suitable for flat-field production, but only if a precision unit costing several thousand dollars is used. Also, such experiments should be conducted in a temperature-controlled room, and after the light source has been stabilized for at least 24 hours. The light from such a source cannot be used to illuminate a flat surface, but rather must be ported into an integrating sphere, which is a hollow sphere with very small ports and a Lambertian inner coating. Besides being very expensive as well, such a sphere is of little use after the CCD sensor is mounted into the camera body.

3.2.3 Solar source

While we were investigating the suitability of other sources, we also tried to obtain a flat-field from a diffuse surface posed near a window. Note that if properly monitored, the short-term temporal uniformity of the sun as a source can be excellent, as sunlight does not flicker. However, we found it generally impossible to reliably duplicate a uniform illumination of a flat diffuse surface. Every time we posed the target (on an artist's easel, mounted to a flat, uniform, and non-reflective surface), the pattern of light distribution varied.

As a location providing no less than 50 degrees of completely unobstructed open sky was available, we began to consider using transmitted sunlight as a flat field. Although the resulting setup could not be as completely specified as if a particular model of xenon lamp, for example, was used, the experiment could still be duplicated anywhere that an expanse of open sky was available. Also, we realized that the spectral content of the illumination could vary without our knowledge and affect the camera's response. The freedom from flicker and the apparent uniformity of a patch of blue sky far from the sun led us to investigate this

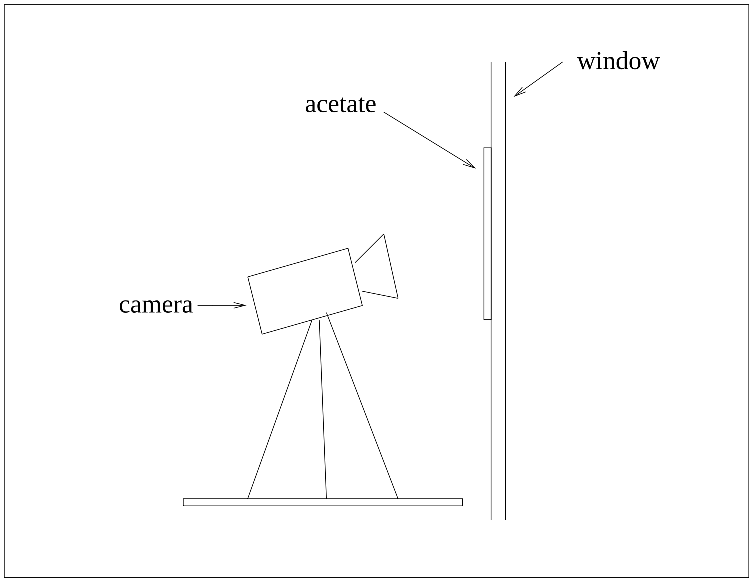


Figure 3.1: The test setup used for capturing flat field images using the solar source.

possibility.

We began by thoroughly cleaning the window inside and out. Then, a double layer of matte acetate was mounted as a diffusing target to increase uniformity and also to reduce the illumination level somewhat. The camera lens was positioned about six inches away from the acetate, pointed at the center of the open expanse, and focused to infinity. This test setup is illustrated in Figure 3.1.

We used a Nikon 58mm f/2.8 Micro-Nikkor lens for most of our experiments, as the performance of this lens was rated at the top in several surveys we found on the Web. The F-mount allowed us to always mount the lens in exactly the same rotational orientation, although we later found the lens to have excellent symmetry of response with respect to the optical axis. In order to obtain varying levels of sensor illumination, we varied the lens aperture (minimum aperture: f/32) and also used high-quality neutral density filters.

We conducted our experiments under clear skies in April 1998 between 10:00 AM and 4:00 PM. We found that after 4:00 PM, the level of illumination began to decrease perceptibly from minute to minute. Images were collected in sets of 100 at no less than 25 fps, so each set was collected in less than 4 seconds. Each run of sets, characterized as a number of sets taken at several different apertures, was collected as quickly as possible (generally within 10 minutes) to provide a tentative basis for comparing the lens response at different apertures.

Uniformity of illumination level within each set of 100 images was verified by taking the average pixel value for each image, identifying the maximum among the 100 averages, and characterizing the other averages as percentages of the maximum. A set, and consequently the entire run, was rejected if a deviation of more than 1 % was discovered for average image values of 100 gray levels or more, on a scale of 0 to 255. For average values of under 100 gray levels, deviation of from 3 to 5 % were sometimes accepted, as we recognized that camera noise processes contributed a greater portion of the error in such images.

Once the uniformity of each set within a run had been verified, each set was condensed into two 1000×1000 32-bit floating-point arrays. The first array was the pixel-by-pixel mean of the 100 images of the set, and the second array was the pixel-by-pixel variance. These two arrays became the input parameters for the model described in Section 6 below.

3.3 Development of pattern noise experimental analysis

In this section, a detailed mathematical analysis of the linear system model is presented. Assumptions are identified and discussed.

We begin by assuming that the behavior of each pixel at any particular moment in time can be described by an equation of the form

$$y = mx + c + \eta = E[y] + \eta \tag{3.1}$$

where

- y is the digital output signal,
- x is the incident illumination (with x_0 defined as zero),
- c is the constant portion of the dark field noise, or additive fixed pattern noise (FPN),
- m is the constant portion of the interpixel photoresponse nonuniformity or multiplicative FPN,
- η is a noise term that includes all other system noises such as shot noise, readout noise, system nonlinearities, and quantization error,
- $E[y]$ is the expected value of y.

The temporal noise η can be modelled as a zero mean Gaussian variable with a variance

$$V[\eta] = V[y] = w_0 + w_1 E[y] \tag{3.2}$$

The term w_0 corresponds to the constant portion of the noise variance, caused mainly by the readout noise, whereas the w_1 term corresponds to the shot noise, which is Poisson distributed with a variance proportional to the output mean. The resultant noise due to these terms can be approximated with a Gaussian distribution.

If the FPN parameters m and c can be determined in advance, the FPN can be compensated prior to further processing. This is likely to improve the performance of the detection algorithm. Also, the temporal noise parameters w_0 and w_1 would help us determine the performance of the algorithm.

Let $x_0, x_1, x_2 ... x_n$ denote a number of intensity values at which observations are made, with $x_0 = 0$ denoting the zero intensity. For each intensity, we can write:

$$y_i = mx_i + c + \eta \tag{3.3}$$

Let the mean and variance of a y_i be denoted by $E_i = E[y_i]$ and $V_i = V[y_i] = \sigma_i^2$. Then,

$$V_i = w_0 + w_1 E_i \tag{3.4}$$

An estimate of the mean and variance can be obtained by using the sample mean and variance of a number of images obtained under identical conditions. A set of such equations obtained by substituting these in (3.4) can be solved in least squares sense to give the values of w_0 and w_1.

For determining values of m and c for each pixel, the following method is used. Denoting the average over N observations with an overbar, we have

$$\bar{y}_i = mx_i + c + \bar{\eta}_i \tag{3.5}$$

Since η_i is assumed to be normally distributed as $N(0, \sigma_i^2)$, $\bar{y}_i$ is normally distributed with parameters

$$E[\bar{y}_i] = E[y_i] = mx_i + c$$
$$V[\bar{y}_i] = V[\bar{\eta}_i] = \frac{V[\eta_i]}{N} = \frac{\sigma_i^2}{N}$$

Consider a neighborhood centered at the current pixel. Assume that the neighborhood is small enough so that the incident illumination x_i remains approximately constant across it, but large enough so that the constituent pixels' nonuniformities will average out to provide us with a good measure of the local incident illumination. Denoting the average value over this neighborhood by the operator μ_s, we have

$$\mu_s[\bar{y}_0] = \mu_s[c] + \mu_s[\bar{\eta}_0]$$

$$\mu_s[\bar{y}_i] = \mu_s[m]x_i + \mu_s[c] + \mu_s[\bar{\eta}_i]$$

giving

$$x_i = \frac{\mu_s[\bar{y}_i] - \mu_s[\bar{y}_0] - \mu_s[\bar{\eta}_i] + \mu_s[\bar{\eta}_0]}{\mu_s[m]}$$

The noise terms η_i, are assumed to be independent and distributed as $N(0, \sigma_i^2)$. We also assume that these terms are uncorrelated in space, i.e. that the values η_i for any pixel are independent of the values for the pixel's neighbors. Then

$$\mu_s[\bar{\eta}_i] \sim N\left(0, \frac{\sigma_i^2}{NN_s}\right),$$

where N_s is the number of pixels in the averaging neighborhood. Because N_s will be on the order of 50-100, the terms $\mu_s[\bar{\eta}_i]$ and $\mu_s[\bar{\eta}_0]$ will be distributed very close to zero, and we may discard them in the derivation.

Note that only the relations between the various illumination levels x_i are important, and not their absolute values. We may therefore choose any convenient scale for our estimating x_i. Setting $\mu_s[m]$ equal to 1, we obtain an estimate of x_i given by:

$$\hat{x}_i = \mu_s[\bar{y}_i] - \mu_s[\bar{y}_0] \tag{3.6}$$

From 3.5 and 3.6 we get

$$\bar{y}_i - \hat{x}_i = (m-1)\hat{x}_i + c + \bar{\eta}_i.$$

We define the following variables:

$$z_0 \equiv c - \mu_s[\bar{y}_0],$$

$$z_1 \equiv m - 1.$$

After substitutions, we obtain the expression

$$\bar{y}_i - \mu_s[\bar{y}_i] = z_1\hat{x}_i + z_0 + \bar{\eta}_i.$$

We may now express the behavior of any particular pixel across various levels of incident illumination with the linear system

$$y = Az + \zeta,$$

where

$$y = [(\bar{y}_0 - \mu_s[\bar{y}_0]),\ (\bar{y}_1 - \mu_s[\bar{y}_1]),\ \cdots,\ (\bar{y}_n - \mu_s[\bar{y}_n])]^T,$$

$$A = \begin{pmatrix} 0 & \hat{x}_1 & \cdots & \hat{x}_n \\ 1 & 1 & \cdots & 1 \end{pmatrix}^T = \begin{pmatrix} a_0 & a_1 & \cdots & a_n \end{pmatrix}^T,$$

$$z = [z_1 z_0]^T,$$

$$\zeta = [\bar{\eta}_0\ \cdots\ \bar{\eta}_n]^T$$

.

We may also assume that the noise terms η_i are uncorrelated in time, so that the error $\zeta = (y - Az)$ is distributed as $N(0, R)$, where

$$R = \frac{1}{N}\,\mathrm{diag}(\sigma_0^2,\ \sigma_1^2,\ \cdots\sigma_n^2).$$

Therefore, we can apply least-squares methods to estimate z, as well as its covariance P.

3.4 Results of noise estimation

In this section, we describe the results of estimating the spatial and the temporal noises. The spatial noise can be reduced by using the estimates of its parameters for every pixel, to compensate it. However, the temporal noise varies from frame to frame, and therefore cannot be reduced by such a method.

3.4.1 Estimation and correction of spatial noise

In this section, results showing significant reduction of pattern noise in time-averaged images are presented. The model's lens-independence is demonstrated. Failure of the model to reduce noise in individual images is also noted.

CCDs are universally reported to be extremely linear devices. Indeed, the basic assumption of our camera model above is that each pixel operates in essentially a linear fashion. In order to test that assumption, we conducted preliminary experiments to determine the camera's response at different flat-field illumination levels. Results of these experiments are shown in Figure 3.2, where each point represents a mean value of the central 1000 × 1000 pixels of the array, and Figure 3.3, where the mean dark value has been subtracted. We varied the illumination level by changing the lens aperture, and assumed for the purpose of these experiments that the illumination thereby changed in perfect powers of 2. Therefore,

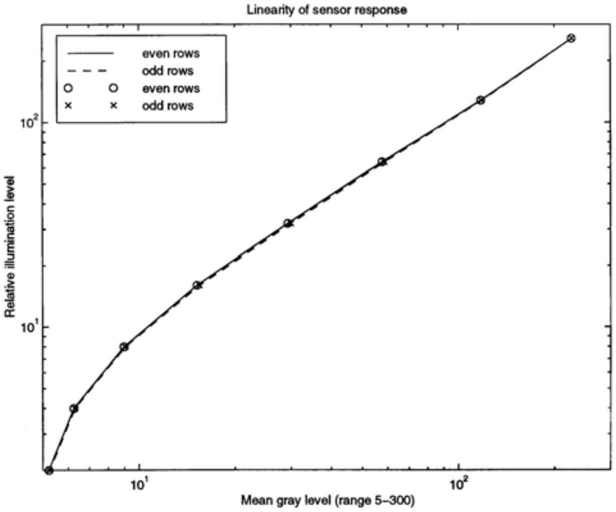


Figure 3.2: Camera's response at different flat-field illumination levels. Each point represents a mean value of the central 1000 × 1000 pixels of the array.

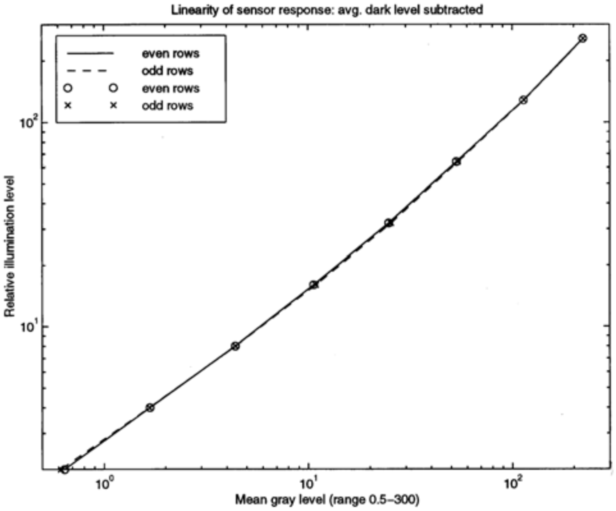


Figure 3.3: Camera's response at different flat-field illumination levels, after subtracting the mean dark value. Each point represents a mean value of the central 1000 × 1000 pixels of the array.

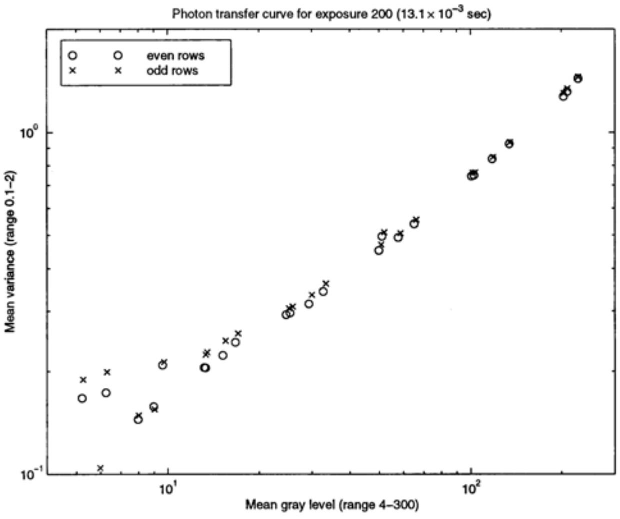


Figure 3.4: The plot of the signal variance against the signal mean.

the largest error in these plots is in the ordinate (illumination level), and not in the abscissa (camera response).

For convention's sake, we also present a rudimentary photon transfer curve in Figures 3.4 and 3.5. This curve, a plot of signal mean against signal variance, is the most commonly used CCD performance curve. Generally, three areas should be discernible, corresponding to the noise process that predominates in each section. To the left, the curve is theoretically flat, as readout noise predominates at low levels. In the center, the curve has a slope of 1, as photon shot noise is the predominant noise process here. On the right, the curve has a slope of 2, corresponding to pixel nonuniformity noise (i.e., fixed pattern noise). Of these three noise processes, of course, only the last is in any way deterministic and compensable. The point where the second and third sections meet represents the signal level at which fixed pattern noise limits the camera's sensitivity. In these terms, the object of this research is essentially to move this point to the right. Our final report will include more refined versions of this curve, updated to incorporate the large amounts of data obtained since these plots were generated.

Our primary lens was the Nikon 58mm Micro-Nikkor, and its response to a high-level flat-field illumination at apertures of f/4 and f/16 is shown in Figures 3.6and 3.7. (Unless noted otherwise, we use examples taken at high levels of illumination throughout this section.

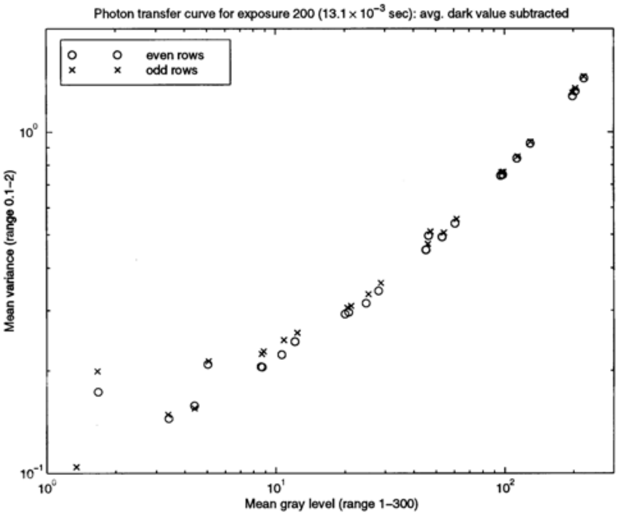


Figure 3.5: The plot of the signal variance against the signal mean after subtracting the mean dark value.

Such images contain the highest noise levels, and also the greatest proportion of compensable noise.) The reduction in lens response as one moves away from the image center is clearly evident. More puzzling to us is the fact that the response becomes more non-uniform as the aperture becomes smaller, as lenses are generally assumed to perform better at smaller apertures. This effect is illustrated in Figure 3.8 by taking a cross-section of the image (i.e., the center row) at different apertures. Efforts to consult with a local optical expert in order to explain this effect are ongoing.

Figure 3.9 shows cross-sectional results at aperture f/4 for several different runs of images (collected as described in section 5 above, over a period of days or weeks), each at a different level of incident illumination. In Figure 3.10, these curves are normalized to correspond to Run #1 at the central pixel. One can see that errors as great as 1 % as evident on the left side of the image. As we would expect the lens response at any one point to be perfectly linear with respect to changing illumination levels, we must assume that these errors represent a flaw in our flat-field illumination. One possible cause is that (with reference to Figure 3.1) our tests were conducted in a room with white walls, and reflections from the room onto the acetate and thereby into the camera may have corrupted our results. Note, however, that the errors are generally much smaller than 1 %, and that the largest normalization factor is nearly 2. Overall, then, this figure demonstrates that our flat-field setup is quite

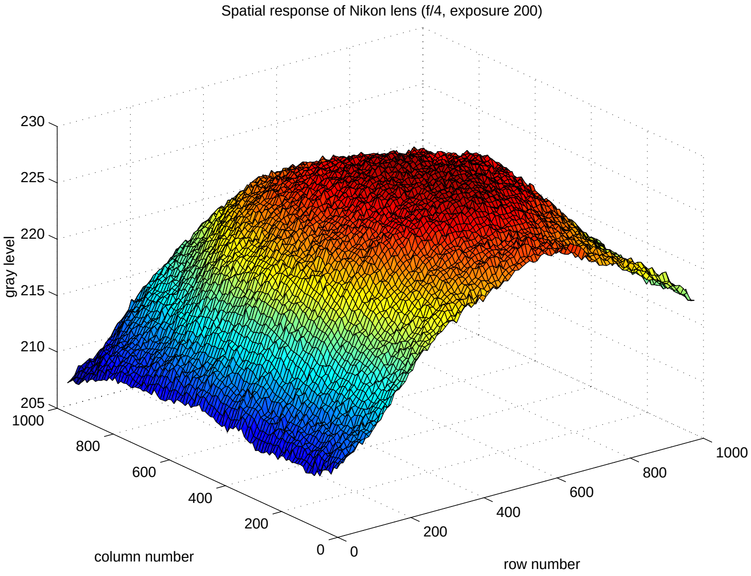


Figure 3.6: The response of the Nikon lens to a high-level flat-field illumination at aperture of f/4.

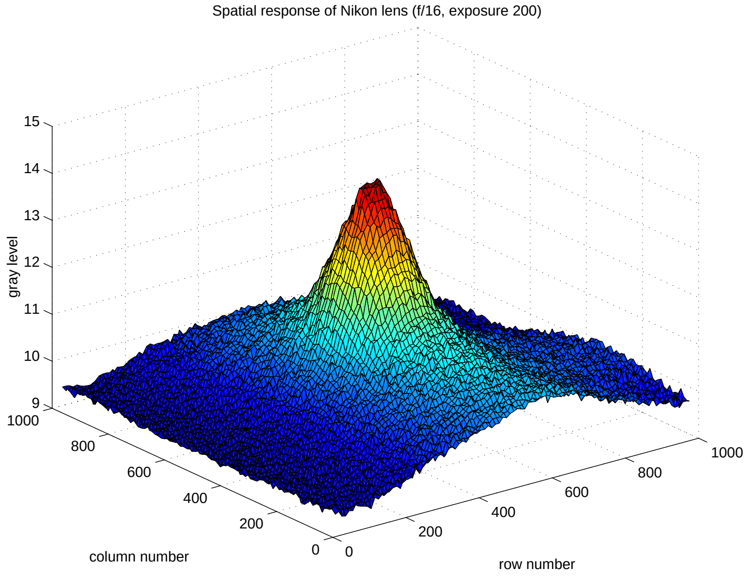


Figure 3.7: The response of the Nikon lens to a high-level flat-field illumination at aperture of f/16.

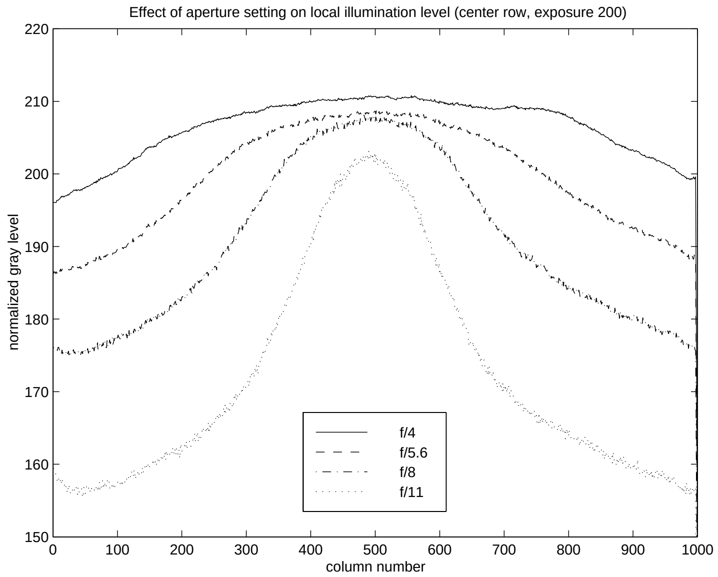


Figure 3.8: The cross section of the lens response at the center row, showing plots of the normalized gray level, for a number of aperture settings.

consistent. We would expect that results could be further improved by posing a black matte shield around the camera to keep out reflections.

One of the runs was arbitrarily selected, and the procedure described in Section 6 above was used to develop a camera noise model. Figure 3.11 shows the result of applying that model to the image of the highest-level set in this run. Essentially, then, we are applying the noise model to itself here. Therefore, this image represents the limit to the amount of noise reduction we can expect. Assuming that the image presented by the lens is a smoother version of the final curve, we can see that a very small level of noise can be expected.

Figure 3.12 shows the result of applying the noise model to the high-level set image of a different run (i.e., one taken on a different day). It is immediately obvious that our model gives a significant degree of improvement.

Figure 3.13 shows the model as applied to a low-level set image of a different run. The level of improvement is much reduced, as the level of compensable (i.e., deterministic) noise in such an image is negligible.

As our model is specific to the array, and not to any particular lens, we also tested its application to set images from a different lens. Figures 3.14 and 3.15 show the flat-field response of a Fujinon zoom lens set to 12.5 mm at apertures of f/5.6 and f/16. It is

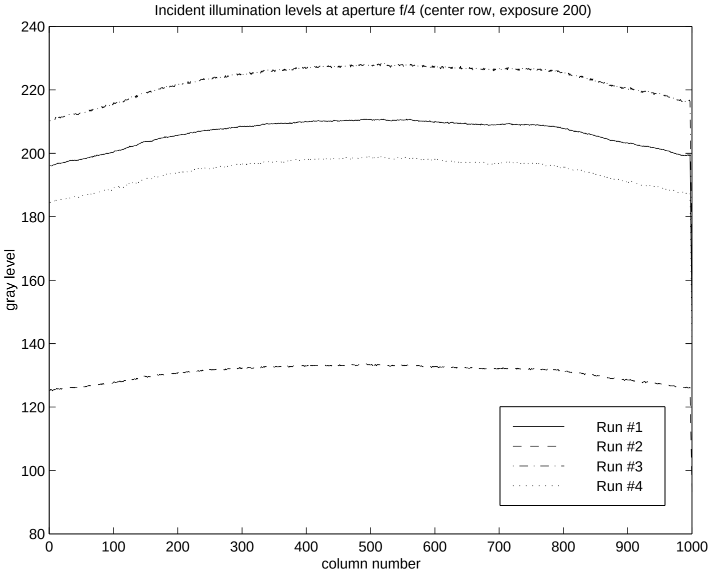


Figure 3.9: Cross sectional results at aperture of f/4 for several different runs of images, each at a different level of illumination.

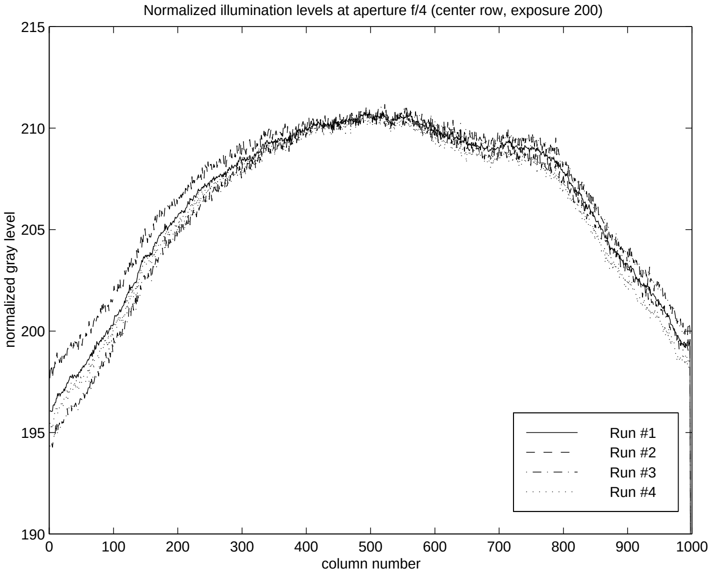


Figure 3.10: Cross sectional results at aperture of f/4 for several different runs of images, each at a different level of illumination, normalized with respect to Run #1 at the central pixel.

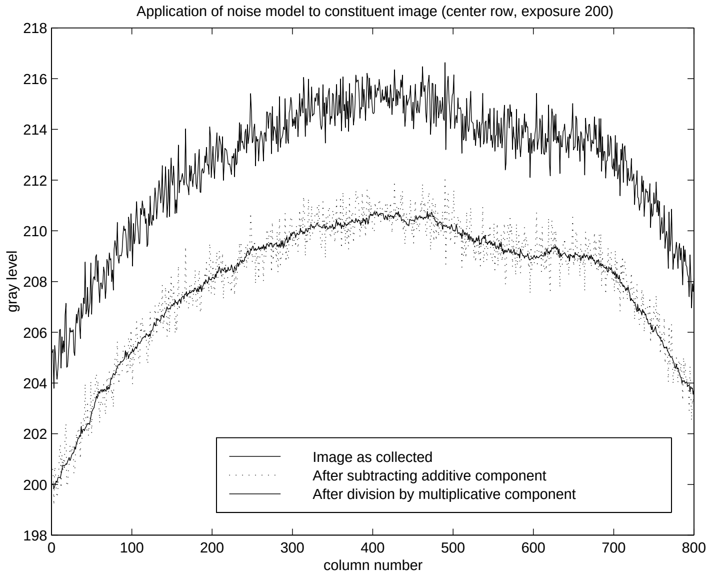


Figure 3.11: Result of applying the camera noise model to an image from a high-level set.

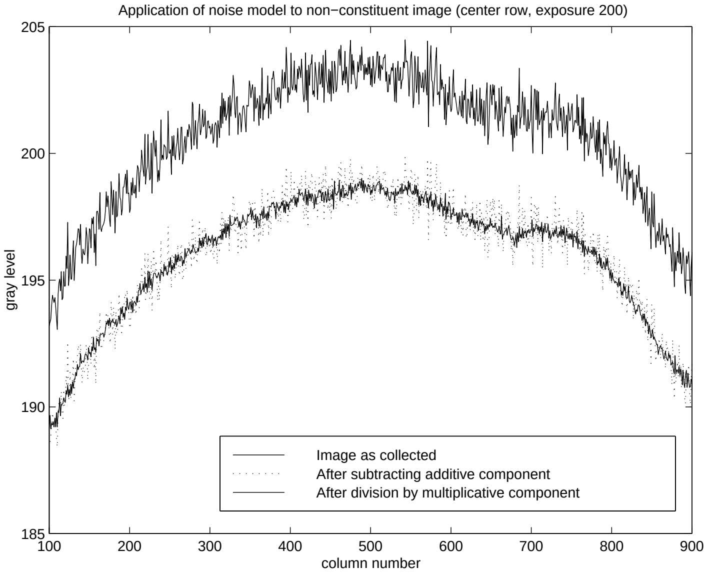


Figure 3.12: Result of applying the camera noise model to an image from a high-level set of a different run.

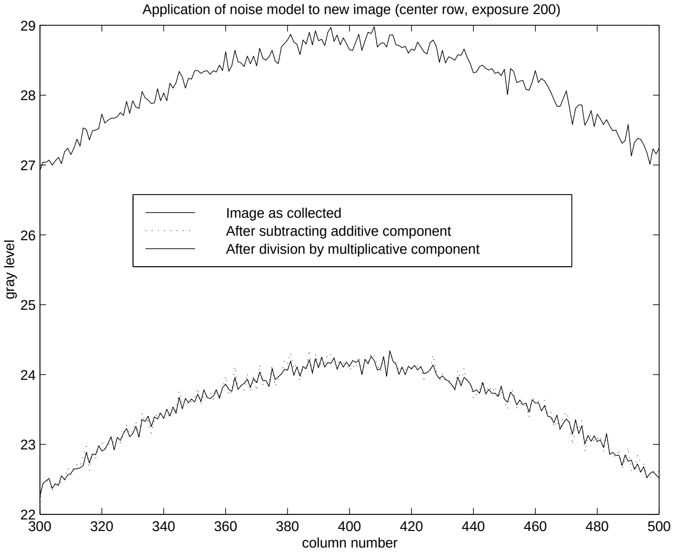


Figure 3.13: Result of applying the camera noise model to an image from a low-level set of a different run.

immediately apparent that the response of this lens is not so precise as that of the Nikon lens, perhaps due to the mechanical compromises necessary for the zoom operation.

Figure 3.16 shows the result of applying our model to a high-level set from this lens. Comparison to Figure 3.12 shows that many noise spikes occur in the same location, as would be expected. However, the general shape of the curve shows the same dip to the right, suggesting again that our flat-field is slightly flawed. Improvement does not quite reach the level of that in Figure 3.12, but is excellent nonetheless.

Finally, we note that in all of the results above, the noise model was applied not to individually captured images, but rather to the mean image taken from 100 individual images. Therefore, temporal (nondeterministic) noise was significantly reduced. In Figure 3.17, we applied the noise model to an individual image. One can see that our model had little effect beyond smoothing the quantized levels. The results above clearly demonstrate that for a set of time-averaged images, our model can offer a significant reduction in camera-generated noise. Unfortunately, for individually captured images, temporal noise processes predominate which cannot be removed.

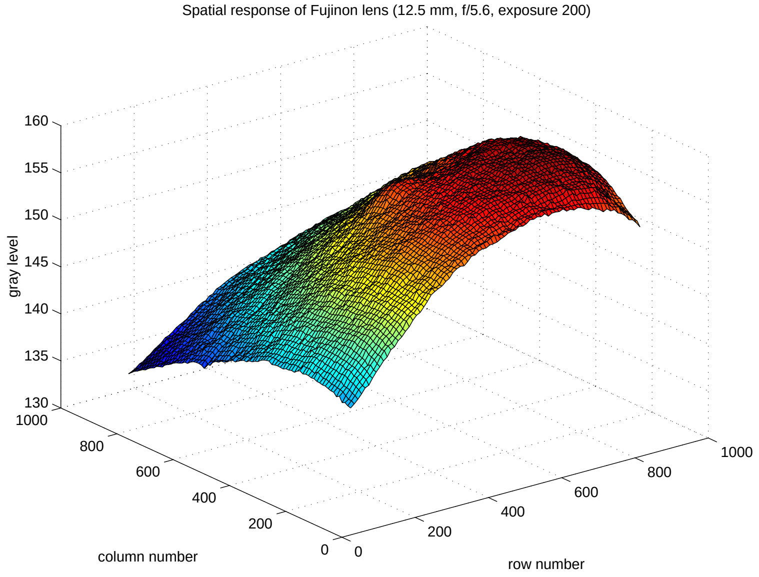


Figure 3.14: Flat-field response of a Fujinon zoom lens set to 12.5 mm at f/5.6 aperture.

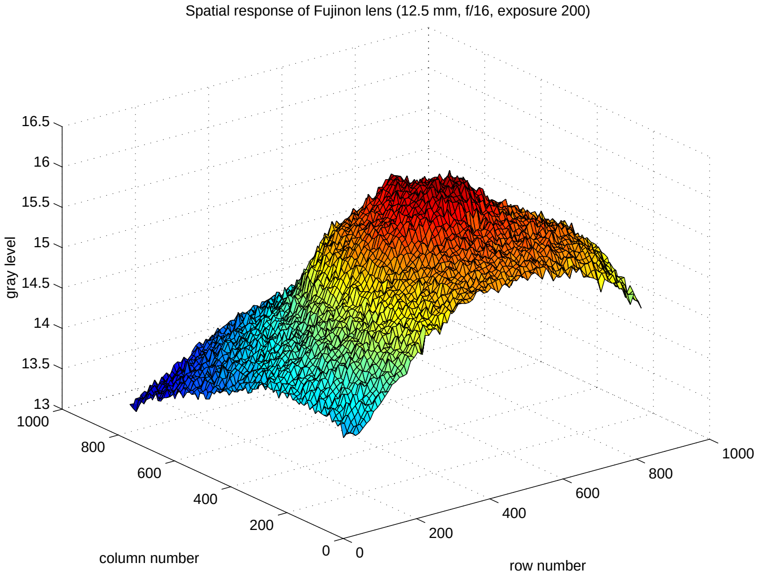


Figure 3.15: Flat-field response of a Fujinon zoom lens set to 12.5 mm at f/16 aperture.

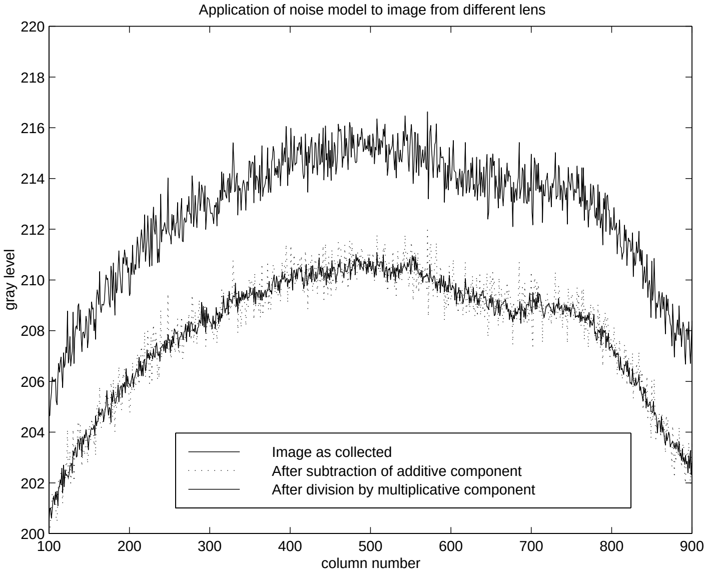


Figure 3.16: Result of applying our model to a high-level set from the Fujinon lens.

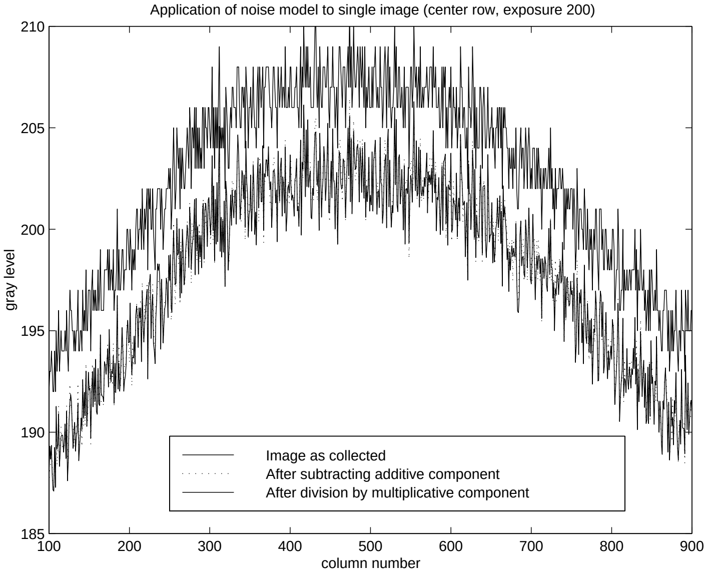


Figure 3.17: Result of applying our model on an individual image, instead of the mean of images. The model has little effect beyond smoothing the quantized levels.

3.4.2 Estimation of temporal noise

To estimate the temporal noise in the camera images, we collected runs of sets of images. Each set contained 100 images (i.e. $N = 100$), collected at a rate greater than 28 fps (i.e. over a period of less than 4 seconds). In order to verify that the illumination level was constant across this period, we took the image-wide mean of the central 1000×1000 pixel region of each image, and found the difference between the means of the brightest and darkest images in the set (relative to the mean of the darkest image). For brightly-lit images (i.e. mean gray level from 150 to 255), the maximum such difference we accepted was 1.02%; most differences were less than 0.5%. At lower levels of illumination, the increasing proportion of temporal noise caused this difference to rise, but even at the lowest illumination levels we accepted no runs with differences greater than 5%. Each run comprised three to eight sets of images, taken at progressively narrower aperture settings. We collected a run within as short a time as possible, to permit us to assume if necessary that the available illumination had remained constant (such an assumption is not necessary for the model described here).

Each accepted set of images was later condensed into two floating-point arrays of size 1024×1024, representing the sample mean and sample variance values at each pixel.

For estimating the temporal noise parameters, imagewide means of these arrays are used as estimates of E_i and V_i in the equation:

$$V_i = w_0 + w_1 E_i \tag{3.7}$$

The equation is solved using least squares to obtain the paramters w_0 and w_1. The plot of V_i to E_i is shown in Fig. 3.18. Its slope is w_1 and the y-intercept is w_0. The values of these parameters obtained are:

$$w_0 = 0.171, w_1 = 5.6 \times 10^{-3} \tag{3.8}$$

This corresponds to a noise variance of $\sigma^2 = 0.888$ or standard deviation of $\sigma = 0.942$ for the background value of 128. This value is used in experiments for testing target detection algorithms.

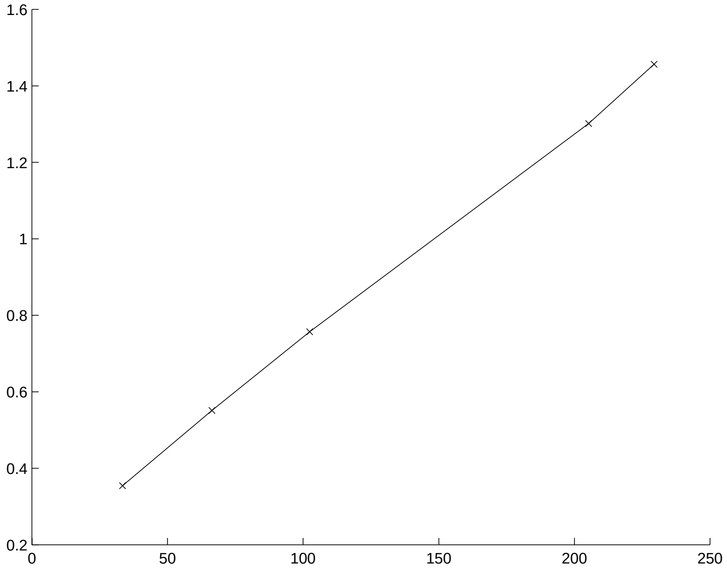


Figure 3.18: The plot of the sample variance V_i against the sample mean E_i for different values of E_i. It is seen that the plot is approximately a straight line, from which the parameters of the temporal noise can be obtained

Chapter 4

Future Work

The following avenues of future work can be explored. The spatial frequency response of the optical system consisting of the lens and the camera, can be characterized using the concept of Modulation Transfer Function (MTF) as described below. Also, the camera noise models can be validated using a statistical hypothesis test.

4.1 Description of Modulation Transfer Function estimation

The spatial frequency response of an optical system is commonly characterized with a plot of the system's modulation transfer function (or MTF), which shows the normalized magnitude of the system's response to a range of input frequencies. The MTF may be expressed as the Fourier transform of the system's line spread function (LSF), which is the response to a flat-field containing a single sharp line. In other words, the LSF is a two-dimensional analog to an impulse response. (For the purposes of this overview, we will ignore the fact that a CCD array responds differently in the row and the column dimensions.)

Traditionally, the LSF of a CCD is obtained by projecting a very narrow band of light onto the array. In an SPIE paper, Lin and Chan describe a method of computing the MTF from the edge-spread function, which may be differentiated to obtain the LSF [4]. The edge-spread function is obtained from a high-contrast target, making the measurement more flexible, more accurate, and less expensive than the traditional method of LSF measurement.

4.2 Noise model validation

An important issue to be addressed as part of this research is the validity of the noise models used to generate synthetic images. In other words, one must answer the question of whether the computer generated images are indeed a set of representative images suitable for the performance characterization of the detection algorithms. This can be done by using a statistical hypothesis test described in [3].

Bibliography

[1] G. E. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, March 1994.

[2] G. C. Holst. *CCD Arrays, Cameras and Displays.* JCD Publishing, Winter Park, FL, 1996.

[3] T. Kanungo, R. M. Haralick, H. S. Baird, W. Stutzle, and D. Madigan. Document degradation models: Parameter estimation and model validation. In *IAPR Workshop on Machine Vision and Applications*, pages 552–557, Kawasaki, Japan, 1994.

[4] S.-Y. Lin and W. H. Chan. MTF and CTF measuring system for digital still cameras. In *Proceedings of the SPIE*, volume 3019, pages 50–57.

[5] D. Wood. *Jane's World Aircraft Recognition Handbook.* Jane's Information Group Ltd., Coulsdon, UK, 1992.

Part II
Detection Algorithms

Abstract

A sequence of images contains more information than a single image. Due to this reason, image sequence analysis has been used in computer vision for quite some time. In particular, a sequence of images is useful for object detection when the camera moves relative to the object. Due to the relative motion, objects at different distances from the camera have different image motion. Using this property, one can obtain information on structure of the 3-D scene as well as the relative motion between the camera and the scene. Furthermore, the individual images are corrupted by camera noise. Use of a sequence of images enables suppression of this noise for reliable detection of low contrast objects.

A computer vision based system that can aid the pilot to detect obstacles in the flight path of an aircraft can be useful for avoiding collisions. Such a system would also be useful for development of a Synthetic Vision System (SVS) proposed for use in a High Speed Civil Transport (HSCT) aircraft with limited cockpit visibility. For this purpose, we had implemented a number of algorithms to detect airborne obstacles using image sequences obtained from a camera mounted on an aircraft. The performance of these algorithms was characterized in presence of camera noise using theoretical and experimental methods. Since the performance degrades in the presence of background clutter, a special approach to address the problem of hazard detection in presence of clutter was studied. This approach uses the differences in the behavior of translation and expansion of image features corresponding to the objects on a collision course and the background clutter. Algorithm fusion for combining different algorithms to overcome their individual limitations was also studied.

In addition to this work on detecting objects on collision course, algorithms for detecting objects crossing the aircraft were designed and implemented on a real-time system. The image processing and tracking steps of the system are described in this part, whereas the hardware implementation is described in the next part.

Chapter 1

Introduction

Image sequence analysis has been widely used in computer vision. This part describes the use of image sequences for detection of airborne obstacles in the flight path of an aircraft.

Continued advances in the fields of image processing and computer vision have raised interest in their suitability to aid pilots to detect possible obstacles in their flight paths. For the last few years, NASA has been exploring the use of image sequences for detecting obstacles in the flight path of an aircraft. NASA Langley Research Center supported a project to enable pilots to 'see through fog' using Passive Milli-Meter Wave (PMMW) images of low resolution. For this project, Tang and Devadiga [12] from our group had developed methods to locate the runway and detect obstacles on and outside the runway. The resulting output can be used by the pilots to decide whether to land or not.

Obstacle detection is also possible with visible-light image sequences. In the design of a High Speed Civil Transport (HSCT) aircraft with a limited cockpit visibility, NASA has proposed a Synthetic Vision System (SVS) in which high resolution video images would be obtained using cameras mounted on the aircraft. These images can be used to detect obstacles in the flight path to warn the pilots and avoid collisions. For aircraft operations, both airborne obstacles, as well as the obstacles on the runway surface should be detected.

Algorithms for detection of airborne objects from images are abundant in the published literature. A systematic performance characterization of a number of target detection algorithms was performed by using image degradation models for digital cameras. It was observed that the algorithms that were studied have a good performance on images which do not have background clutter. However, the performance degrades severely when background clutter is present. Thus, the goal of this work has been to design algorithms which perform better in cluttered background environments, with low probabilities of false alarms and mis-detections and capability of target detection early enough to avoid a possible collision. To achieve this goal, a special approach was used to discriminate hazardous objects on collision course from the background clutter. Algorithm fusion was studied for combining different algorithms in a statistical framework, to overcome their individual limitations. The performance of the fused algorithm was found to be better than the individual algorithms under appropriate conditions.

This part of the report is organized as follows: Chapter 2 describes the basic, well-known algorithms used for detection of airborne obstacles. These algorithms were tested on real image sequences provided by NASA. In Chapter 3, the performance of these algorithms is experimentally characterized using the approach described by Kanungo et al. [10]. The theoretical characterization of the algorithms' performance is described in Chapter 4, and the experimental performance is compared with the theoretical performance.

The main contribution of the research for the detection of hazardous objects is described in the next two chapters. A special approach is proposed for discrimination of objects on collision or near-collision course from background clutter. This approach is described in Chapter 5 where differences in the behavior of translation and expansion in the image are used to separate hazardous objects from clutter. Chapter 6 describes the Bayesian methodology used for combining detection algorithms in a statistical framework. Performance of fused algorithm is compared with that of the individual algorithms.

In addition to hazardous objects, it is also useful to detect and track objects crossing in front of the aircraft. A real-time system using pipelined image processing hardware was designed for this purpose. Chapter 7 describes the image processing operations which are performed by the pipelined hardware, and the tracking operations performed on the host machine to form a complete real-time system.

Chapter 8 concludes the part and explores avenues for future work.

Chapter 2

Object Detection Algorithms

This chapter describes the algorithms that were implemented to detect airborne obstacles in the flight path of a flying aircraft. Statistical theory used for target detection is first described, followed by a number of basic steps useful for removing background clutter, amplifying the signal to noise ratio, and detecting objects having different sizes and velocities. Results obtained by using real image sequences are also described.

2.1 Background

NASA's need for enhanced capabilities in obstacle detection using image processing requires robust, reliable and fast techniques. These techniques should provide a high probability of detection while maintaining a low probability of false alarm in noisy, cluttered images of possible targets, exhibiting a wide range of complexities. The size of the image target can be quite small, from sub-pixel to a few pixels in size. As an example, consider a Cessna aircraft that has a length and wing-span of approximately $9\,m$ $(30\,ft)$ and the fuselage diameter of approximately $1.2\,m$ $(4\,ft)$ [19]. The detection algorithm must be capable of detecting this small target at least 25 seconds prior to a possible collision to allow for corrective actions by the pilot. Assuming that both the aircraft are traveling at $125\,m/s$ (250 knots), their relative velocity can be as high as $250\,m/s$ (500 knots). In such case, they would be $6.25\,km$ (3.5 nautical miles) apart 25 seconds before collision. Using a camera with a resolution of 60 pixels per degree, the image size of the aircraft is 5.0×0.7 pixels from a side view, but only 0.7×0.7 pixels from a front view. Furthermore, the detection algorithm must report such targets in a timely fashion, imposing severe constraints on their execution time. Finally, the system must not only work well under the controlled conditions found in a laboratory and with data closely matching the hypothesis used in the design process, but it must be insensitive – i.e., must be *robust* – to data uncertainty due to various sources, including sensor noise, weather conditions, and cluttered backgrounds.

Extensive work has been done on the problem of target detection. When the signal to noise ratio is low, it is preferable to use the 'track before detect' approach. In this approach, an object is tracked over multiple frames before making a hard decision on the presence or absence of a target. The simplest way to integrate the input images over multiple frames is by temporally averaging them. When the image motion of the object is very small, as in the case of an object being exactly on a collision course [14], this happens to be the best approach. However, if the object has a significant image motion, other approaches are needed. Nishiguchi et al. [16] proposed the use of a recursive algorithm to integrate multiple frames while accounting for small object motion. A dynamic programming approach was used by Barniv [4] and Arnold et al. [2] to detect moving objects of small size. The theoretical performance of this approach was characterized by Tonissen and Evans [18].

The above algorithms perform well when the background is uniform. However, in real situations the hazardous object should be detected not only against uniform background, but also against backgrounds such as clouds, ground or water. The features introduced due to a non-uniform background which interfere with object detection are collectively known as clutter. Thus, the objective of the detection algorithms is to successfully detect the hazardous object, without giving unnecessary false alarms from clutter. Subtraction of consecutive images is often used to remove stationary clutter. However, an object on a collision course could be nearly stationary in the image [14]. Hence, this method is not useful for our application, since it could remove the object as well. Alternatively, morphological filtering [6] removes objects of large size, usually corresponding to clutter while retaining the objects of small size. This approach is useful in removing large clutter, such as clouds. But it does not remove small-sized clutter.

2.2 Statistical decision theory for target detection

Statistical decision theory [13, 17] can be used to design optimal or near-optimal detection algorithms, as well as to characterize their performance. The input to the algorithm is a sequence of images, each composed of a large number of individual pixels. These pixels are degraded by various sources, such as atmosphere, lens, and camera noise. Based on the statistical behavior of this degradation, the image pixels can be combined in space and time, to make statistically optimal decision about the presence or absence of a target. For making these decisions, probabilistic models of the signal and its degradation can be used.

Let H_0 and H_1 denote the hypotheses that the target is absent or present, respectively, and $P(H_0)$ and $P(H_1)$ denote their respective prior probabilities. Let z represent the vector of observations from which one is supposed to determine the presence or absence of a target. By Bayes' rule, the posterior probabilities are given by:

$$P(H_1|z) = \frac{p(z|H_1)P(H_1)}{p(z)}\ ,\ P(H_0|z) = \frac{p(z|H_0)p(H_0)}{p(z)} \tag{2.1}$$

The ratio of these probabilities is given by:

$$\frac{P(H_1|z)}{P(H_0|z)} = \frac{P(H_1)p(z|H_1)}{P(H_0)p(z|H_0)} = \frac{P(H_1)}{P(H_0)}L_H(z) \tag{2.2}$$

where $L_H(z)$ proportional to the ratio of the probabilities is called the likelihood ratio.

When the algorithm reports a target even where there actually is none, it is called a false alarm, whereas when it does not report an existing target, it is called a mis-detection. The performance of a detection algorithm is characterized in terms of false alarms and mis-detections. According to the Neyman Pearson criterion [13, 17], the number of mis-detections for a given rate of false alarms can be minimized by thresholding the likelihood ratio $L_H(z)$. The threshold is a function of the required rate of false alarms. In place of the likelihood ratio, any of its monotonic function (such as the logarithm) can be used. Such a function is called a discriminant function.

To decrease the probabilities of false alarms and mis-detections, one can integrate observations spatially or temporally. Let the N elements $z_1, z_2 \ldots z_N$ of z be independent observations. The likelihood ratio and its logarithm (log likelihood ratio) are given by:

$$L_H(z) = \frac{p(z_1, z_2 \ldots z_N|H_1)}{p(z_1, z_2 \ldots z_N|H_0)} = \prod_{i=1}^{N} \frac{p(z_i|H_1)}{p(z_i|H_0)} \tag{2.3}$$

$$l(z) = \log L_H(z) = \sum_{i=1}^{N} [\log p(z_i|H_1) - \log p(z_i|H_0)] \tag{2.4}$$

In the case of z_i's having normal distributions in absence and presence of target, such that their probability density functions are:

$$p(z_i|H_0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{z_i^2}{2\sigma^2}\right]\ ,\ p(z_i|H_1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{(z_i-\mu)^2}{2\sigma^2}\right] \tag{2.5}$$

The log likelihood ratio is given by:

$$l(z) = \log L_H(z) = \sum_{i=1}^{N} \frac{-(z_i-\mu)^2 + z_i^2}{2\sigma^2} = \frac{\mu}{\sigma^2}\left[\sum_{i=1}^{N} z_i\right] - \frac{N\mu^2}{2\sigma^2} \tag{2.6}$$

This is a monotonic function of $\sum z_i$. Hence, thresholding the sum (or mean) of the observations yields an optimal detector. Since sum and mean are linear functions, they are also normally distributed.

Consider a discriminant function which is normally distributed in absence and presence of target as $N(\mu_0, \sigma_0^2)$ and $N(\mu_1, \sigma_1^2)$, respectively with equal variances $\sigma_0^2 = \sigma_1^2$ but unequal means μ_0 and μ_1. If this function is thresholded to obtain a particular false alarm rate, it can be shown that the corresponding mis-detection rate is a function of its Signal to Noise Ratio (SNR) given by $(\mu_1 - \mu_0)/\sigma_0$. Hence in this case, the performance in terms of false alarm and mis-detection rates is determined by the SNR.

If N independent normal observations are made, their sum is distributed as $N(0, N\sigma^2)$ in absence of target, and $N(N\mu, N\sigma^2)$ in presence of target. Hence, the SNR is given by $N\mu/\sqrt{N\sigma^2} = \sqrt{N}\mu/\sigma$ – i.e., amplified by a factor of $\sqrt{N}$. In other words, a signal with SNR of $S/\sqrt{N}$ integrated over N frames could yield the same rate of false alarms and mis-detections as one would get using a single observation with SNR of S. Hence, the SNR *required* for detection *reduces* by $\sqrt{N}$ when N frames are added. The same result is true for averaging of N frames, since the signal as well as the noise would be reduced by a factor of N.

2.3 Pre-processing

Before any other algorithms can be applied, pre-processing should be performed on the input images to suppress the background. The following approaches were used for pre-processing the images.

2.3.1 Low-stop filter

In the case of an image with little or no clutter, a low-stop filter which subtracts from every pixel, the local average of the neighborhood of that pixel effectively suppresses the background intensity. This filter can be implemented by convolving the image with a 2-D mask corresponding to the filter. Since the amount of computation increases with the mask size, a small sized mask was used in conjunction with the pyramid approach described in Section 2.4 to simulate the effect of a large sized mask.

2.3.2 Morphological filter

If the background has significant clutter, the low-stop filter is not as effective for removing it. A morphological filter [6] can remove large sized features (usually clutter), while retaining small sized features (usually targets).

The gray-scale morphological operations of dilation ($\oplus$) and erosion ($\ominus$) are defined as:

$$\begin{aligned}(f \oplus m)(x,y) &= \max_{(x',y')\in m} \{f(x-x', y-y') + m(x',y')\} \qquad (2.7)\\ (f \ominus m)(x,y) &= \min_{(x',y')\in m} \{f(x+x', y+y') - m(x',y')\} \qquad (2.8)\end{aligned}$$

where m is the mask using which the morphological operation is performed, and f is the image which is considered to have a default value of $-\infty$ outside its domain. Morphological closing and opening can be defined using the above operations as:

$$\begin{aligned}(f \bullet m) &= (f \oplus m) \ominus m \geq f \qquad (2.9)\\ (f \circ m) &= (f \ominus m) \oplus m \leq f \qquad (2.10)\end{aligned}$$

A difference between the original image and its morphological opening, known as the top-hat transform outputs small-sized positive targets – i.e., bright targets in dark background. On the other hand, the difference between the morphological closing and the original image, known as the bottom-hat transform outputs negative targets – i.e., dark targets in bright background. Each of these images are non-negative, and can be separately used to detect targets.

A single mask for these morphological operations gives undesirable outputs for jagged boundaries of large features. Hence, horizontal mask m_x and vertical mask m_y were used separately as proposed by [6]. These masks are of length 5 with origin at the center of the mask, with all the pixels having the default value of zero. The outputs are given by:

$$\begin{aligned}F_+ &= F - \max\{F \circ m_x, F \circ m_y\} \qquad (2.11)\\ F_- &= -F + \min\{F \bullet m_x, F \bullet m_y\} \qquad (2.12)\end{aligned}$$

2.4 Spatial integration

To detect targets of a number of different sizes and velocities, and to amplify the SNR, the target pixels in a given image can be integrated by forming an image pyramid. For this purpose, the following basic operations are used:

1. Low-pass filter (LP or $\overline{LP}$): Convolves the image in x and y directions with the masks $m_x = m_y = [1, 3, (3), 1]/8$, or their mirror images. The parentheses denote the origins of the masks.

$$\begin{aligned} f_{LP}(x,y) &= \sum_{x'}\sum_{y'} f(x-x', y-y')\, m_x(x')\, m_y(y') \\ f_{\overline{LP}}(x,y) &= \sum_{x'}\sum_{y'} f(x+x', y+y')\, m_x(x')\, m_y(y') \end{aligned} \tag{2.13}$$

2. Down-sampler (DS): Selects even numbered pixels in the input image to give an image with half the resolution.

$$f_{DS}(x,y) = f(2x, 2y) \tag{2.14}$$

3. Up-sampler (US): Forms the output image by putting the input image pixels in even numbered positions, and zeros in odd numbered positions. The image is scaled by 2 to maintain the image intensity during subsequent low-pass filter step.

$$f_{US}(x,y) = 2f(x/2, y/2) \text{ when } x,\ y \text{ are even; 0 otherwise} \tag{2.15}$$

These steps are combined to form two types of operations:

1. Low-pass down-sample operation ($LP \to DS$): Decreases the resolution of the image by two. Low-pass filter prevents aliasing of high frequencies in the image by suppressing them.

2. Up-sample low-pass operation ($US \to \overline{LP}$): Increases the resolution of the image by two. Low-pass filter smoothes the output of the up-sampler (containing zeros at odd pixels) to produce the effect of interpolation. In this case, the mirror image masks are used to compensate the asymmetry in the masks.

The above operations can be used to combine pyramid formation with low stop or morphological filtering by using the system shown in Figure 2.1. Images $pyr[i]$ are formed by successively applying low-pass and down-sample operations on the original image. These images can be directly used as inputs to the morphological filter to detect targets at different resolutions. Images $pyr'[i]$ are formed by successively applying up-sample and low-pass operations to the lowest resolution image $pyr[n]$, where n is the number of pyramid levels. These operations remove the high frequency components of the original image. Low-stop filtered images are given by $ls[i] = pyr[i] - pyr'[i]$, and retain only the higher frequency components not subtracted out by $pyr'[i]$.

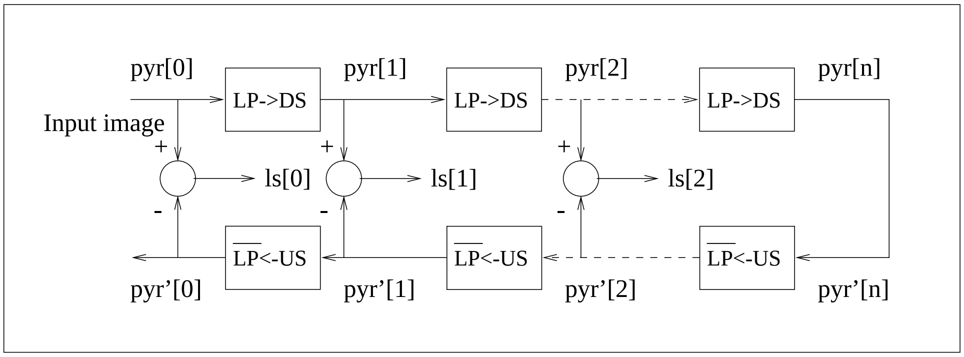


Figure 2.1: Spatial integration using pyramid construction: LP or $\overline{LP}$: low-pass filtering with original mask or its mirror image, DS: down-sampling, US: up-sampling. The pyramid images at stage $i = 0 \ldots n$ are denoted by $pyr[i]$. Low-stop filtered images are obtained the by subtracting the corresponding up-sampled pyramid outputs $pyr'[i]$ from $pyr[i]$ and are denoted by $ls[i]$.

In this way, a hierarchy of images, each with half the resolution of the previous one is formed. The size as well as the velocity of the object in the image scales as the resolution is lowered. There is a particular resolution at which the object occupies no more than 2 to 3 pixels in length and width, which would be optimal for detection of the object.

2.5 Temporal integration

As shown in Section 2.2, integration of pixels corresponding to a target results in amplification of the target SNR, and increased reliability of detection. Depending on the image motion of the target, the following approaches can be used for integration of target pixels over a number of image frames. The performance of these approaches is characterized experimentally and theoretically in Chapters 3 and 4, respectively.

2.5.1 Recursive temporal averaging

In the case of objects on a collision course [14] the image motion is very small. Hence, pixel wise temporal averaging of a sequence of images would improve the detection performance. However, direct use of temporal averaging results in infinite memory. To give a higher weight to more recent observations, a recursive filter can be used. The output $F(k)$ at time k for any pixel is recursively obtained from the input $f(k)$ at the same pixel using the following steps:

1. Initialization: $F(0) = 0$

2. Recursion: $F(k) = f(k) + \alpha F(k-1)$

where α is a forgetting factor between 0 (full forgetting) and 1 (no forgetting).

2.5.2 Dynamic programming

In the case of moving targets, the temporal averaging filter does not improve the detection. A dynamic programming algorithm [2] is more effective in detection of moving targets. The algorithm is based on shifting the images before averaging them so as to align the target to be detected. Since the velocity of the target could be arbitrary, the velocity space (u, v) is discretized within the range of possible target velocities. A set of intermediate images F, each corresponding to a particular velocity (u, v), are created recursively using the following steps:

1. Initialization: For all pixels (x, y) and all velocities (u, v), set

$$F(x, y; u, v; 0) = 0 \tag{2.16}$$

2. Recursion: At time k, set

$$F(x, y; u, v; k) = f(x, y; k) + \alpha \max_{(x',y') \in Q} F(x - u - x', y - v - y'; u, v; k-1) \tag{2.17}$$

   where

$$Q = \{(x', y') | x'_{min} \leq x' \leq x'_{max} \ , \ y'_{min} \leq y' \leq y'_{max}\} \tag{2.18}$$

3. Termination: At time K, take

$$F_{max}(x, y; K) = \max_{(u,v) \in P} F(x, y; u, v; K) \tag{2.19}$$

   where

$$P = \{(u, v) | u_{min} \leq u \leq u_{max} \ , \ v_{min} \leq v \leq v_{max}\} \tag{2.20}$$

The maximum operation in the recursion step is performed using the set Q, which ensures that the targets with velocities which do not fall on the grid are not missed. The set of discretized velocities denoted by P determines the range of target velocities that can be detected by the algorithm. The final maximum in the termination step combines the targets corresponding to all the velocities. The number of elements in P and Q are denoted by p and q, respectively.

In the recursion step, a maximum is taken over q pixels. If these pixels are all noise pixels, they are more likely to give a false alarm if q is large. Thus, the rate of false alarms increases with q. To get better performance, a smallest possible q should be used. The value of $q = 4$ has been used in our experiments corresponding to a 2×2 neighborhood, given by:

$$Q = \{(0,0), (-1,0), (0,-1), (-1,-1)\} \tag{2.21}$$

This ensures that the targets having fractional velocities are not missed. The asymmetry in this neighborhood is compensated by choosing $u_{min} = u_{max} - 1$ and $v_{min} = v_{max} - 1$. For the case of $u_{max} = v_{max} = 1$, $p = 4$ and P is given by:

$$P = \{(0,0), (1,0), (0,1), (1,1)\} \tag{2.22}$$

The algorithm then detects targets with a maximum velocity of 1 pixel per frame. However, when spatial integration is performed prior to dynamic programming, targets with larger sizes and velocities can be detected.

On the other hand, if $P = Q = \{(0,0)\}$ so that $p = q = 1$, the algorithm reduces to recursive temporal averaging, which gives the best performance for stationary targets. However, the performance of temporal averaging sharply degrades if the target is moving, whereas that of dynamic programming algorithm does not.

The output of the dynamic programming algorithm is an image, with large values at positions where the target strength is high. However, the pixels in the neighborhood of the target will also have a significantly large value. This can be resolved by using non-maximal suppression, where the output is smoothed using a Gaussian filter with $\sigma = 1.0$, and each pixel which is not a local maximum in its 3×3 region is set to zero. After this, only the pixels which are local maxima remain, which can be thresholded to obtain the target locations.

It should be noted that separate processing should be performed if the targets are negative – i.e., dark targets on a bright background. In the case of low-stop pre-processing, this is done by using the negative of the pre-processed image, whereas in the case of morphological pre-processing, both original minus open and closed minus original images are processed separately.

2.6 Composite system

The above mentioned algorithms have been combined to form a composite system for target detection. The steps that form this composite system are:

1. Temporal Averaging: This step is performed first in the case of objects in a uniform background, having a very small image motion, such as those on a collision or near-collision course. In such a case, temporal averaging improves the SNR and reduces the processing rate required for subsequent steps.

2. Pyramid construction with low-stop or morphological filtering: In this step, a pyramid is constructed to accommodate different sizes and velocities of objects. For pre-processing the images, low-stop or morphological filtering is performed at each pyramid level to remove background intensity. Low-stop filtering is more effective in low clutter situations, whereas morphological filtering [6] is more effective in suppressing background clutter due to clouds and ground.

3. Dynamic Programming: A dynamic programming algorithm [2] is performed on pre-processed frames to integrate the signal over a number of frames by taking the target motion into consideration. Non-maximal suppression and thresholding are then performed on the output.

It should be noted that one or more of these steps can be bypassed so that any of the basic algorithms described above can be tested individually using the same system.

2.7 Results using analog camera

The above target detection algorithms were applied to real image sequences obtained from NASA. Figure 2.2 (a) shows an image from the sequence with the target aircraft flying away from the host aircraft. The sequence can be played in reverse to simulate the aircraft on a collision course. Since the aircraft on a collision course have a small image motion, temporal averaging was the optimal detection algorithm in this particular case. The aircraft was at a distance of approximately 4 nautical miles ($7.4\,km$), and was barely visible in a single image. Low-stop filter was applied before temporal averaging to remove the near-uniform background. After temporally averaging and thresholding, the aircraft was detected as shown in Figure 2.2 (b). Dynamic programming algorithm was performed on a sequence of images

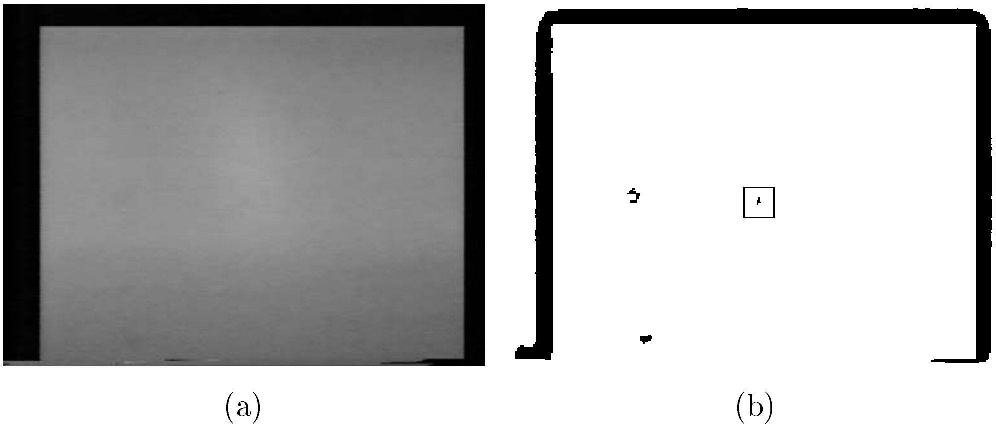
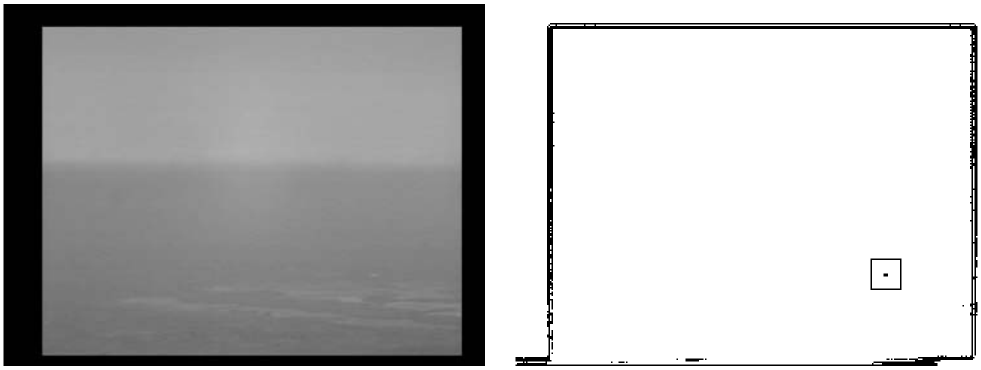


Figure 2.2: Target detection using temporal averaging: (a) Original image with a distant contracting target at 4 nautical miles. The target is approximately in the middle of the image. However, due to degradation of image quality, it is very faint. (b) Detection of the distant contracting target using low-stop filter pre-processing, temporal averaging and thresholding. A false alarm in the mid-left area is most likely due to a smudge on the camera.

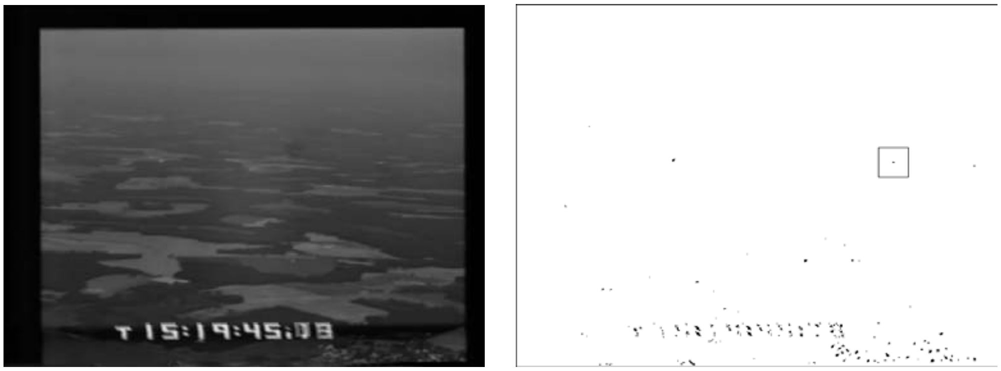
(after applying low-stop filter as pre-processing) in which an aircraft was flying from right to left across the image as shown in Figure 2.3 (a). Dynamic programming algorithm detected the aircraft with a low rate of false alarms. However, the target was dilated by the use of this algorithm. Clutter removal using morphological filtering was also explored. Figure 2.4 (a) shows a small aircraft flying in the middle-right part of the image. The image was actually obtained by averaging 10 motion compensated images from an image sequence, in which an aircraft was flying on the collision course. Application of morphological filter removed most of the clutter due to edges of large-sized features. This aircraft which was on a collision course, was retained. However, other small-sized features were also retained, resulting in a number of false alarms. The result is shown in Figure 2.4 (b).

Chapter 3 presents a systematic performance characterization for temporal averaging as well as dynamic programming using statistical image models for digital cameras. It was observed that the algorithms performed very well when the background was clear. However, the performance degraded severely in presence of clutter. In the case of cluttered images, pre-processing using morphological filter worked better than that using low-stop filter. Most of the clutter was removed, but small sized clutter, especially due to specular reflection from water remained. Finally it was observed that the number of false alarms after applying the



(a) (b)

Figure 2.3: Target detection using dynamic programming: (a) Original image frame with a translating target. (b) Location of the detected target using dynamic programming (following a low-stop pre-processing step).



(a) (b)

Figure 2.4: Detection using morphological processing: (a) An average of ten motion-compensated frames of an image sequence. The aircraft is in the middle-right part of the image. (b) Detection using morphological filter. False alarms due to other features are also seen.

algorithm, in general, was reduced but still significant.

2.8 Data collection using digital camera

The real data that was used for our previous work was captured using an analog camera and recorded using NTSC video, thus containing additional noise that should not be present when a digital camera is used on the actual flight. Hence, the performance of the algorithms should be characterized without the undue interference from video noise. For this purpose, a system was designed to capture image sequences from an aircraft using a digital camera, and record them digitally on a disk. The camera used was $1K \times 1K$ Kodak MegaPlus ES1.0 camera with the output at approximately 30 frames per second and a gray scale resolution of 8 bits. Hence, a bandwidth of 30 MBytes per second and a storage of 108 GBytes per hour of recording is required.

To capture the video image sequences with these large bandwidth and storage requirements, as well as perform the image processing operations in real time, a real-time image processing system with pipelined image processor called DataCube MaxPCI was procured. This system is a cost-effective way to meet high-throughput low-latency demands and has become popular among researchers working on real-time vision problems. The New Technology Disk (NTD) available with the DataCube MaxPCI has the required ability to perform high-speed digital image recording. NTD is a Redundant Array of Inexpensive Disks (RAID) that enables high-speed lossless digital image recording and playback. The image data can be recorded and played back at a real-time frame rate (overall 40 MBytes/sec).

Image data has been obtained from flight tests conducted at NASA Langley Research Center. A sample image captured using this system is shown in Figure 2.5. Work on implementing the detection algorithms on the DataCube hardware using these images is described in [11]. Detection of objects crossing the aircraft (instead of those on a collision course) was performed on the DataCube system in real time. The algorithms used for this purpose are described in Chapter 7.

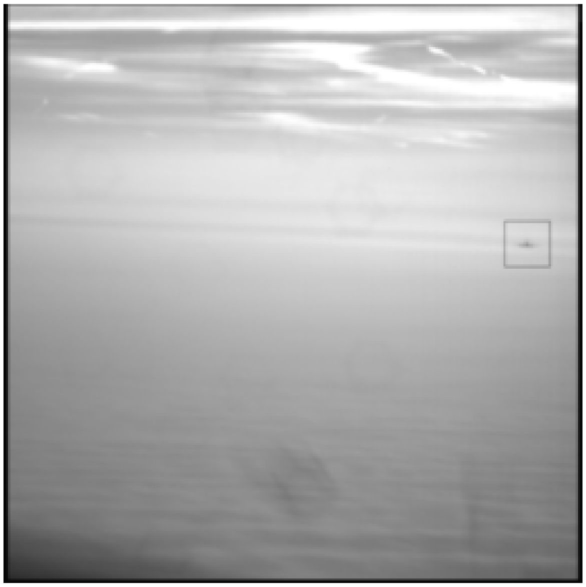


Figure 2.5: An image captured from an aircraft using the digital recording system. The target aircraft is in the middle-right part of the image.

Chapter 3

Performance Characterization of Detection Algorithms

The most common tool used to characterize the performance of a detection algorithm is a plot of its probability of mis-detection versus its probability of false alarm, as some tuning parameter is changed. This plot is commonly known as the “receiver operating curve” of the system, or ROC, for short. Although ROCs are useful to represent the system performance as a parameter is varied, they have several limitations. One disadvantage in using ROCs is due to the fact that only one parameter can be varied at a time. Thus, if the effect of variations of multiple variables needs to be studied, a different curve must be determined for each of these variables making the analysis of the system performance more difficult. A second disadvantage is that it is difficult to compare ROCs for different algorithms since they may take different variables into account. Finally, obtaining ROCs is an expensive process where factorial experiments must be carried out to determine the system performance at all performance levels with the probability of false alarms ranging from zero to one.

In Kanungo et al. [10], a methodology which was adapted from the psychology literature, and is discussed next, was proposed as an alternative characterization tool to summarize multiple ROCs into a single curve, solving the problems described above. This chapter describes how to use this methodology to characterize the performances of the algorithms described in Chapter 2. The performance of the dynamic programming algorithm is compared against that of temporal averaging, and thresholding of a single image frame.

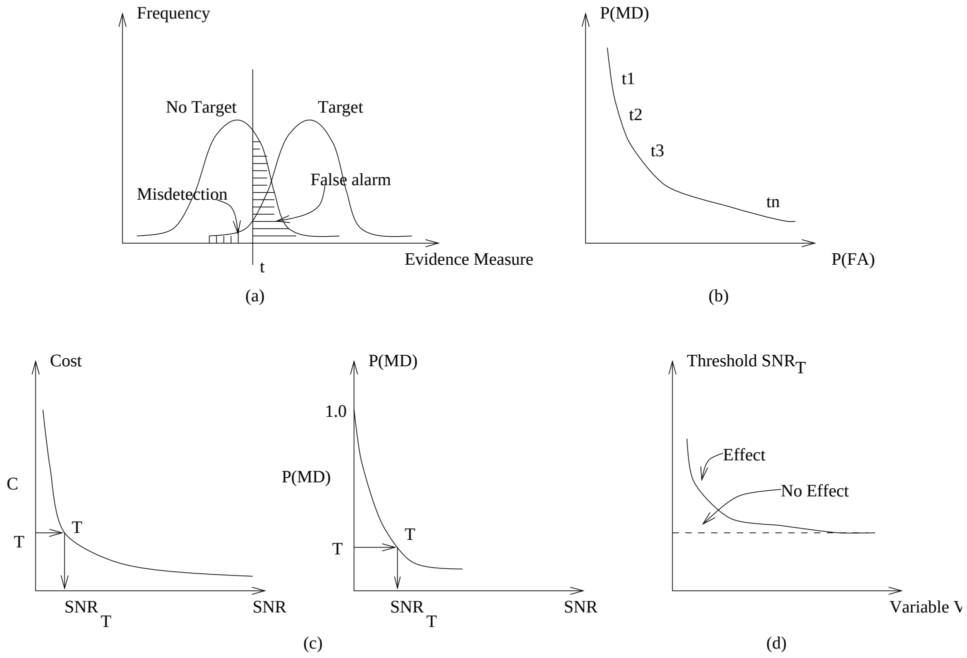


Figure 3.1: Steps for performance characterization: (a) Step 1: Obtain the frequency distributions of the evidence measure for images with and without target. (b) Step 2: Obtain the ROC. (c) Step 3: Determine the optimal operating point using either the expected cost or the probability of detection given the probability of false alarm. (d) Step 4: Plot the threshold value corresponding to the optimal operating point versus a variable of interest.

3.1 Performance characterization methodology

For the sake of completeness, the methodology for performance characterization proposed in [10] is described here. Consider a detection algorithm that must report whether a given image has a target or not. Typically, the algorithm would compute some measure of evidence of target presence and compare it to some given threshold value. Whenever the evidence measure is greater than the given threshold, a target would be reported. The performance of the algorithm is affected by several factors, such as image contrast, target size, complexity of the background, etc. The effect of variations of these variables on the overall performance can be measured through the use of equivalent effects of some critical signal variable by following the four steps described below.

1. Obtain evidence distributions: The first step consists on estimating distributions of evidence measures, one for images with target and another for images without target,

as illustrated in Figure 3.1 (a). This estimation is done non-parametrically by randomly presenting the algorithm with images of both types and recording the frequency of the evidence measure values reported by the algorithm, using a histogram. It should be noted that the frequency distributions are used here only for estimating the false alarm and mis-detection rates. The evidence measure which is thresholded may or may not be derived from these distributions according to Bayes' rule. Hence, the performance of optimal as well as non-optimal detectors can be characterized by this approach.

2. Obtain ROCs: The second step consists on constructing an ROC as the one shown in Figure 3.1 (b) by varying the threshold used by the algorithm to compare against the computed evidence measure. False alarms occur when a pixel in the given image does not contain a target, but the evidence measure is greater than the threshold being used. Mis-detections occur when the given image contains a target, but the evidence measure is less than the threshold. The probabilities of false alarms and mis-detections can be approximated by their frequency ratios:

$$\begin{aligned} P(FA) &= P(H_1|H_0) = \frac{\text{Number of false alarms}}{\text{Total number of input pixels without target}} \\ P(MD) &= P(H_0|H_1) = \frac{\text{Number of mis-detections}}{\text{Total number of targets in input images}} \end{aligned}$$

   where H_0 and H_1 denote the hypotheses corresponding to the absence and presence of a target, respectively.

3. Determining the optimal operating point: The optimal operating point (or its corresponding threshold value) can be specified in different ways, depending on how much prior knowledge is available. If the prior probabilities and costs are known, the optimal operating point can be defined as the one minimizing the expected cost. Let C_{10}, C_{01}, C_{11}, and C_{00}, be the costs of a false alarm, a mis-detection, a correct detection, and a correct rejection, respectively. The expected cost is then given by:

$$\begin{aligned} E[C] &= [P(H_0|H_0)C_{00} + P(H_1|H_0)C_{10}]\,P(H_0) \\ &+ [P(H_0|H_1)C_{01} + P(H_0|H_1)C_{11}]\,P(H_1) \end{aligned} \tag{3.1}$$

   The optimal operating point is found by minimizing $E[C]$ with respect to the threshold to be used by the algorithm. In the most likely case when the costs are difficult to set, an alternative way to define the required operating point is to use the Neyman-Pearson criterion – i.e., to maximize the probability of detection for a *given* probability of false alarm.

Independently of which definition is used, the optimal operating point depends on the signal to noise ratio (SNR) in the input image. For example, increasing the target contrast results in an increase of the SNR and, hopefully, in an improvement of the algorithm performance for a given threshold value. The optimal operating points for different SNRs can be found by repeating steps 1 and 2 for the corresponding SNR values and determining the optimal point for each of the resulting ROCs. Once this is done, a graph of the expected cost or the probability of detection versus SNR can be plotted, depending on which definition of operating point is being used. This is illustrated in Figure 3.1(c). Finally, let SNR_T and T be the SNR and the associated threshold values for the optimal operating point for a given level of performance, as shown in the figure. The level of performance is specified by either a desired expected cost of classification or a desired probability of mis-detection, again, depending on which optimal criterion is used.

4. Performance analysis with respect to variables of interest: Besides SNR, other factors affect the algorithm performance and merit study. Examples are the size of the target, the amount of target motion on the images, and the amount and nature of image clutter. In order to study these effects, steps 1 to 3 are repeated for different values of variables representing these variations. These results are then summarized in a graph where the threshold T determined in step 3 is plotted against the value of the variable of interest, as shown in Figure 3.1(d). A fairly flat plot indicates that the effect of the variable being considered on the optimal operating point of the algorithm is negligible. On the other hand, a steep plot indicates that the variable has a high impact on the performance.

It should be noted that a smaller SNR threshold T implies better performance, since weaker targets can be detected with the same given rates of false alarms and mis-detections. Measuring the performance in terms of the SNR threshold makes it easier to measure and compare the performance of different algorithms, or the same algorithm with different parameters. This is because the variables, such as the false alarm and mis-detection rates are eliminated from the curves, making place for other parameters.

3.2 Experimental protocol

In this section, the experimental protocol used to characterize the performance of the target detection algorithms, is described in detail. The protocol consists of the following components, specifying how to

1. Generate images of simulated targets,
2. Apply the detection algorithm,
3. Estimate the rates of false alarms and mis-detections (ROCs) for different sets of parameters, and
4. Characterize the algorithm performance by condensing the ROCs into a performance curve.

3.2.1 Image generation

In order to characterize the performance of the detection algorithm, it is applied to sequences of synthetic images with and without targets. While the images with targets are used to estimate the mis-detection rate, the images without targets are used to estimate the false alarm rate. The images can have the following different types of backgrounds:

1. Synthetic noise from camera model: The background is assumed to have a constant value A_{bg}. The noise is artificially simulated, using the camera noise model.
2. Real noise from a digital camera: The background images are taken from a sequence of images obtained from a digital camera looking at a scene with constant intensity such as clear sky, or white paper.
3. Real background an from analog camera: The background images are obtained using a sequence of images with significant clutter. The sequence, which was provided by NASA, was captured using an analog camera mounted on a flying aircraft. Figure 3.2 shows a typical frame of this sequence.

Generation of image sequences

To estimate the number of false alarms, the background images themselves, without any addition of targets are used directly. The size of these images is $N_x \times N_y$. For estimation of

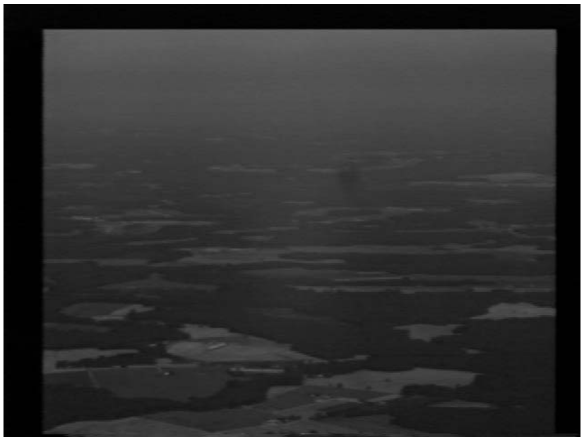


Figure 3.2: A sample image from the real background sequence provided by NASA. The image sequence was taken from an analog camera mounted on an aircraft.

the rate of mis-detections, simulated targets are inserted in the background images generated as described below. For each simulation, a target file is created having information on the position, velocity, size, amplitude and each target to be placed in an image. The image size is taken as $N_x \times N_y$. The number of targets to be inserted in every image is N_{targ}. The target trajectories are generated in such a way that the detection of one target does not interfere with the detection of another. This is accomplished by drawing a window around each target trajectory. The next generated trajectory is valid only if the window around it does not overlap with the windows around the previously generated targets. Otherwise, the procedure is repeated by generating another trajectory, until the total number of valid trajectories is N_{targ}.

The velocity (V_x, V_y) of the targets is uniformly distributed so that $-u_{max} \leq V_x \leq u_{max}$ and $-v_{max} \leq V_y \leq v_{max}$. The position of the targets is specified for the *last* frame – i.e. when the detection is completed. The position of the target in other frames is given by $(x - V_x \Delta t, y - V_y \Delta t)$, where Δt is the time-interval between the given frame and the last frame.

A target can be a point target, or have a specified height and width. The size of the target is given by $s_x \times s_y$. The target amplitude is given by A. For point targets, the amplitude corresponds to the contrast of the pixel it occupies, with respect to the background. However, for an extended target, the contrasts of all the occupied pixels are given by the product of the target amplitude and the fraction of the area in the respective pixel that is covered by the target.

Figure 3.3 (a) shows the trajectories of simulated targets to be added to an image, and Figure 3.3 (b) shows a zoomed part on a portion of the image. The end of the trajectories are marked by blobs. The black box around the target denotes the region where another target cannot be present, to reduce the interference between the targets.

Once the file describing the targets is created, an image sequence of N_{frame} frames is generated. For each frame, the position of the targets are calculated, and the targets are inserted accordingly. For point targets, the amplitude is added to the background image in the target position pixel. For extended targets occupying a number of pixels (fully or partially), the product of the amplitude and the fractional occupancy is added to the background image at that pixel.

Addition of noise

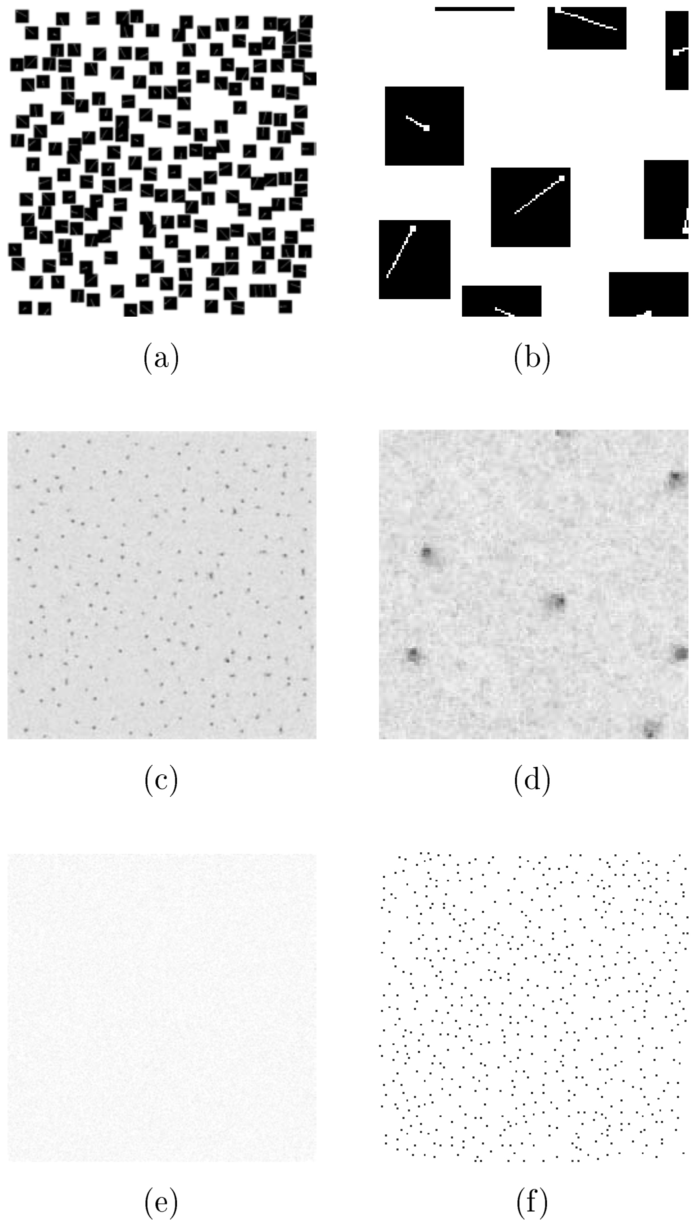
Two types of camera noise [8, 11], the Fixed Pattern Noise (FPN) and the temporal noise are added to the sequences created using synthetic backgrounds. FPN has two components, additive and multiplicative. The parameters of this noise change from pixel to pixel, but do not change with time. The parameter values for each pixel are determined a priori using the camera, and stored as images. On the other hand, the temporal noise is completely random, and is generated separately for each frame. The temporal noise approximately follows a Gaussian distribution with a variance of:

$$\sigma_{noise}^2 = w_0 + w_1 I$$

where I is the expected gray value of the pixel, and w_0, w_1 are the parameters of the particular camera. However, since the background amplitude A_{bg} is constant for the experiments with simulated noise, and the target amplitude $A \ll A_{bg}$, we have $I = A + A_{bg} \simeq A_{bg}$ and σ_{noise}^2 is approximately constant, given by:

$$\sigma_{noise}^2 \simeq w_0 + w_1 A_{bg}$$

Hence, the noise can be approximated as Gaussian noise with a constant standard deviation of σ_{noise}. The values of the parameters for the particular camera were estimated [11] as $w_0 = 0.171$ and $w_1 = 0.0056$. For background $A = 128$, this gives $\sigma_{noise} = 0.942$. The image is quantized to give the output in byte format.


(a) (b)

(c) (d)

(e) (f)

Figure 3.3: Detection using dynamic programming: (a) Simulated targets trajectories. There are 200 targets, and the image size is 960 × 960. The end of the trajectory is marked by a blob. The targets are separated so that there the interference between them is reduced. The black box around the target denotes the region where another target cannot be present. (b) A zoomed part of the target trajectory image. (c) The dynamic programming output of a typical experiment (before non-maximal suppression). (d) Zoomed part of the output. (e) The dynamic programming output of the same experiment without adding targets – i.e., false alarms. (f) Zoomed part of the output.

3.2.2 Algorithm application

The target detection algorithm whose performance is to be characterized is applied to each simulated image sequence. In the cases of synthetic images, and digital camera sequences, fixed pattern noise (FPN) can be corrected in advance by using pre-computed parameters of FPN for each pixel. However, these parameters are perturbed by a random amount corresponding to their estimated covariance, to model the error in estimating these values. Experiments are performed without and with correction of FPN, and the results are compared.

According to the type of background used, preprocessing in the form of a low-stop filter or a morphological filter are performed before applying dynamic programming. After dynamic programming is applied, non-maximal suppression is performed to ensure correct counting of false alarms and mis-detections. The output (before non-maximal suppression) of a typical experiment with 200 targets is shown in Figure 3.3 (c) and (d) where the latter shows a zoomed part of the output.

3.2.3 Estimation of false alarms (FA) and mis-detections (MD)

The algorithm to be characterized is applied on the image sequences with as well as without targets. The sequences without targets are used to estimate the false alarm rate, whereas the sequences with targets are used to estimate the mis-detection rate.

For the false alarm rate, the histogram of the output image is obtained. Using this histogram, the false alarm rates for different thresholds can be obtained. For the mis-detection rate, only the pixels in a specified window of 5×5 pixels around the specified target position are checked. For each such window corresponding to a single target, the maximum value of the algorithm output is taken. A histogram of these maximum values is formed, and processed to obtain the mis-detection rates for different thresholds. The false alarm and mis-detection rates are averaged over a number of simulations N_{FA} and N_{MD}, respectively.

The number of simulations to test can be specified so that the standard deviation in the estimate of the false alarm or mis-detection rate is below a given value. This can be seen by observing that the occurrence of an event such as a false alarm or a mis-detection can be modeled as a Poisson process and therefore the variance of the total number of events is equal to the mean. Thus, if n events are observed, the standard deviation of the absolute error in the number of events is $\sqrt{n}$, and that of the relative error is $1/\sqrt{n}$. For example, for

$n = 10$ events, the error σ is 3.2, or 32 % of the number of events. This error estimate can be confirmed by measuring the variance of these rates across the simulations.

3.2.4 Performance characterization

Using the estimated false alarm and mis-detection rates, the receiver operating curve (ROC) can be plotted showing the rate of mis-detection against the rate of false alarms. The mis-detection rate for a specified false alarm rate (FA_T) is noted from the curve. The simulations are repeated for a number of signal amplitudes A. The ratio of this amplitude to noise level corresponds to the SNR. The value of the signal amplitude for a specified mis-detection rate (MD_T), and the above false alarm rate is obtained. This is considered as the threshold signal value (A_T). The number of simulations used is at least $N_{FA} = 10/FA_T$ in the case of false alarms and $N_{MD} = 10/MD_T$ in the case of mis-detections, so that for the rates FA_T and MD_T, an average of at least 10 events would be observed, giving an error σ of at most 32 %. Due to constraints on the execution time, larger number of experiments were not used, although they would be desirable for reducing this error.

Other parameters, such as the size of the target, can be varied one at a time, and the variation of A_T can be plotted against the respective parameter to determine the effect of the parameter on the algorithm performance.

3.3 Results

The target detection algorithm was tested on 3 categories of images as described in the protocol. The results are shown and compared in the following sections.

3.3.1 Synthetic noise from camera model

In this case, the noise was synthetically generated using the noise model of the Kodak Megaplus ES 1.0 digital camera. Targets of varying size were added for mis-detection analysis. Experiments without and with correction of FPN were performed.

Figure 3.4 (a) and (b) show the plots of the false alarm and mis-detection rates, respectively, against the threshold value, for experiments without FPN correction. The mis-detection rates are shown for a number of signal amplitudes for 1 × 1 targets. The mis-detection rate is measured as the ratio of the average number of mis-detections, to the total number of targets in a simulation. However, the false alarm rate is measured as the average

Table 3.1: Table of parameters used for the experiments with the following image categories: (1) Synthetic noise from camera model, (2) Real noise from a digital camera, (3) Real background from an analog camera.

Description	Parameter	Category		
		(1)	(2)	(3)
Image x size	N_x	960	960	640
Image y size	N_y	960	960	480
No. of targets	N_{targ}	200	200	50
Maximum x velocity	u_{max}	1	1	1
Maximum y velocity	v_{max}	1	1	1
x size	s_x	0.5 to 2	2	2
y size	s_y	0.5 to 2	2	2
Amplitude	A	1.0 to 15.0	1.0 to 6.0	10.0 to 70.0
Number of frames	N_{frame}	32	32	32
Background value	A_{bg}	128	$\simeq$ 200	not used
Noise standard deviation	σ_{noise}	0.942	not used	not used
Forgetting factor	α	15/16	15/16	15/16
Number of FA simulations	N_{FA}	500	1	1
Number of MD simulations	N_{MD}	50	10	10
Threshold FA rate	FA_T	0.02	10	10
Threshold MD rate	MD_T	0.001	0.01	0.01

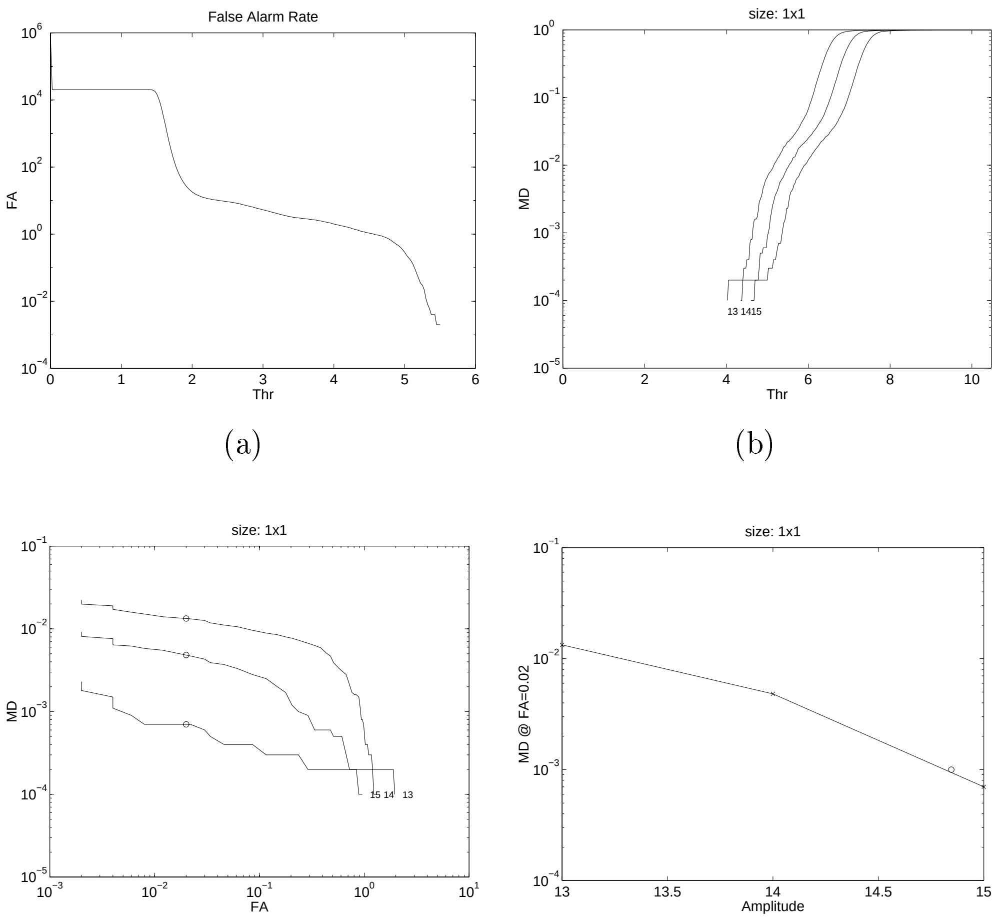
number of false alarms per simulation, instead of the ratio of the number of false alarms to the total number of pixels. This is done to give a better idea of the algorithm performance.

Figure 3.4 (c) shows the plot of mis-detection rate against false alarm rate for different amplitude values for 1×1 targets. The point of threshold false alarm rate FA_T is set to 0.02 false alarms per simulation, which corresponds to a total of 10 false alarms for $N_{FA} = 500$ simulations. Figure 3.4 (d) shows the plot of mis-detection rate against the amplitude values for the above rate of false alarms. The A_T for the threshold mis-detection rate of MD_T is interpolated, and marked as a circle. The MD_T is set to a probability of 0.001 per target, which corresponds to an average of 0.2 mis-detections per simulation for a simulation with 200 targets, or a total of 10 mis-detections for $N_{MD} = 50$ simulations. The corresponding graphs for the case where fixed pattern noise compensation was applied are shown in Figure 3.5.

The above experiments are repeated for other sizes of targets, and the A_T calculated from these is plotted against the size of the target. Resulting plots for the experiments without FPN correction are shown in Figure 3.6 (a) for square targets (size $x \times x$) and in Figure 3.6 (b) for rectangular targets (size $1 \times x$). The corresponding results for the experiments with FPN correction are shown in Figure 3.6 (c) and (d). The threshold amplitudes for various sizes are tabulated in Table 3.2. It is seen that larger targets require smaller signal amplitudes for detection implying better performance. Similarly, the signal amplitudes required when FPN correction is applied are much smaller than those when the correction is not applied, implying better performance in the former case.

Table 3.2: Results of dynamic programming algorithm on simulated image sequences without and with FPN correction. Threshold amplitudes are shown for false alarm rate of 0.02 per simulation and mis-detection rate of 0.001 per target.

Size	No FPN correction	With FPN correction
1×1	14.85	4.72
1×1.5	11.43	3.48
1×2	9.38	3.10
1.5×1.5	10.49	2.55
2×2	6.35	2.04



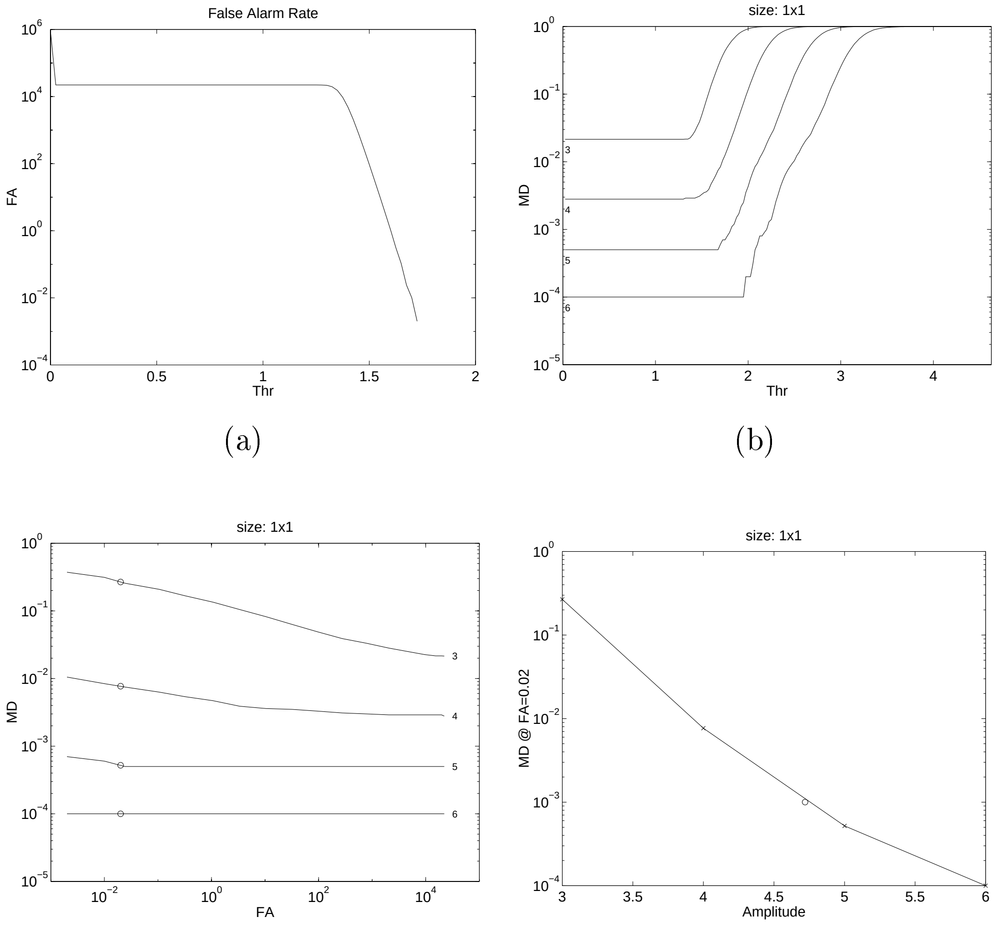
(a)

(b)

(c)

(d)

Figure 3.4: Results for camera noise model without FPN correction: (a) Plot of FA rate (average number per simulation) against threshold (b) Plot of MD rate against threshold, for a number of signal amplitudes (higher amplitudes towards right) for 1 × 1 targets. (c) Plot of MD rate against FA rate (for marked amplitude). The data points are marked as crosses. The MD rate when FA rate is $FA_T = 0.02$ per simulation is interpolated, and plotted as circle. (d) Plot of MD rate against amplitude for FA rate of $FA_T = 0.02$ per simulation. The value amplitude when MD rate is $MD_T = 0.001$ per target is interpolated and marked as a circle.



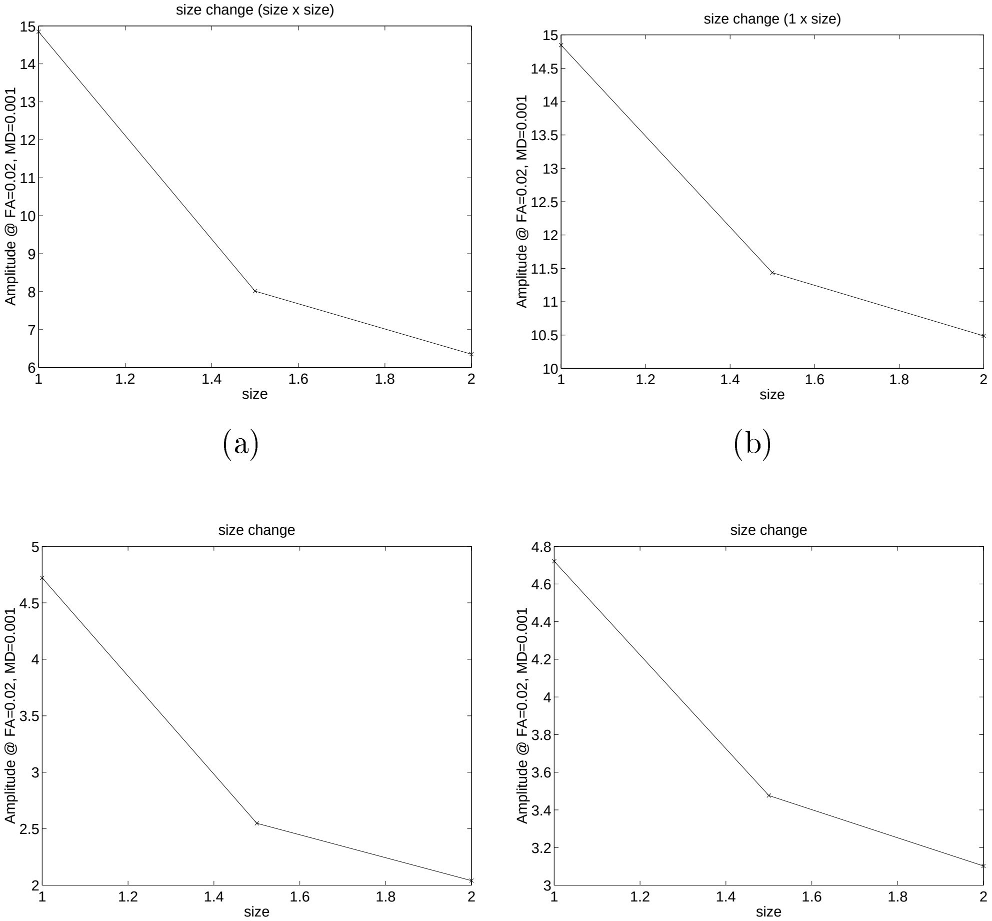
(a)

(b)

(c)

(d)

Figure 3.5: Results for camera noise model with FPN correction: (a) Plot of FA rate (average number per simulation) against threshold (b) Plot of MD rate against threshold, for a number of signal amplitudes (higher amplitudes towards right) for 1×1 targets. (c) Plot of MD rate against FA rate (for marked amplitude). The data points are marked as crosses. The MD rate when FA rate is $FA_T = 0.02$ per simulation is interpolated, and plotted as circle. (d) Plot of MD rate against amplitude for FA rate of $FA_T = 0.02$ per simulation. The value amplitude when MD rate is $MD_T = 0.001$ per target is interpolated and marked as a circle.



(a)

(b)

(c)

(d)

Figure 3.6: Performance curves for simulated targets: (a) Plot of amplitude against the target size ($x \times x$) for experiments without FPN correction. The data points are marked as crosses. (b) Plot of amplitude against the target size ($1 \times x$). (c) and (d) Corresponding plots for experiments with FPN correction.

3.3.2 Real noise from a digital camera

In this case, instead of synthetically generating the noise, background images captured using the Kodak Megaplus ES 1.0 digital camera looking at the sky were used. Targets of size 2×2 pixels were synthetically added for the mis-detection analysis. Experiments without and with correction of FPN were also performed.

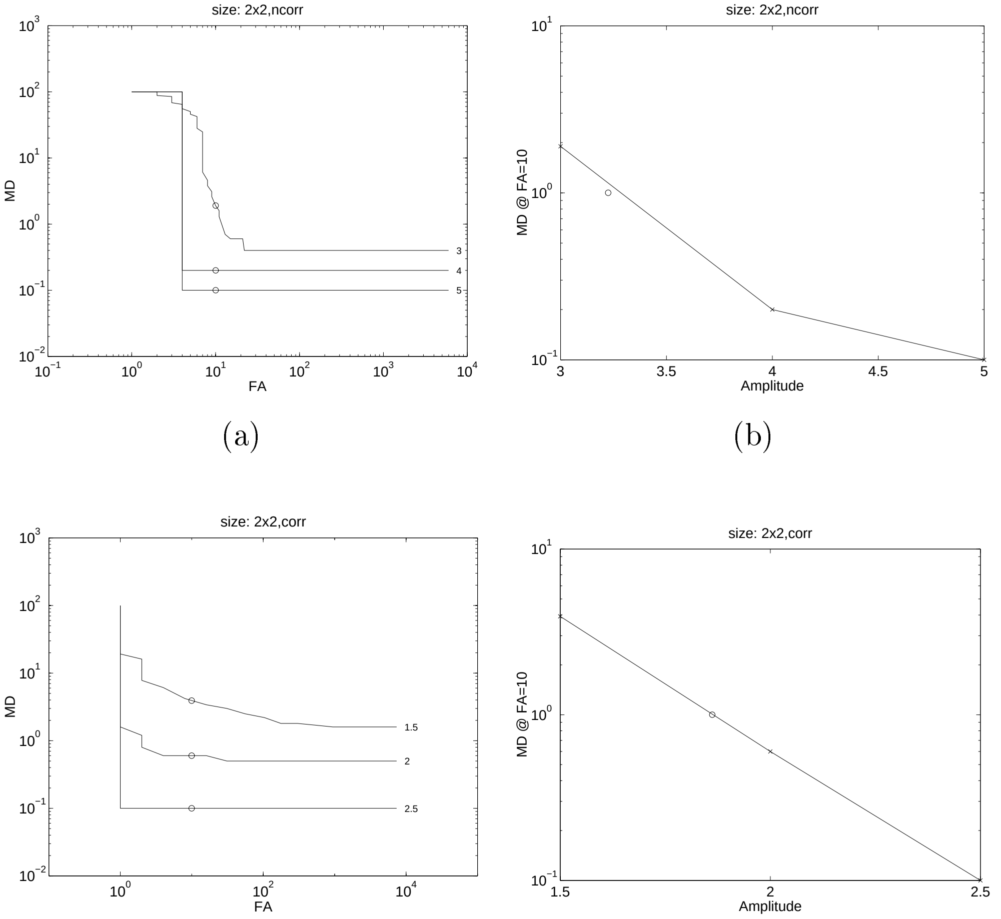
The false alarm threshold was set $FA_T = 10$ per simulation, resulting in a total of 10 false alarms for $N_{FA} = 1$ simulation. The mis-detection threshold was set to $MD_T = 0.01$ per target, corresponding to 20 mis-detections for $N_{MD} = 10$ simulations with $N_{targ} = 200$ targets. Unfortunately, the performance at lower rates of false alarms and mis-detections could not be reliably estimated because of the limited number of background images available. However, one can extrapolate the false alarm and mis-detection rates to study the behavior of the algorithm for lower rates. Due to the normal distribution of noise, even a small increase in the threshold reduces the false alarm and mis-detection rates dramatically. Hence, a somewhat higher target amplitude can be expected to reduce these rates to an acceptable level.

In the case of the experiments without FPN correction, the plot of mis-detection rate against false alarm rate for different levels of target amplitude is shown in Figure 3.7 (a). The plot of mis-detection rate against SNR for false alarm rate of $FA_T = 10$ per simulation is shown in Figure 3.7 (b). The corresponding plots for the experiments with FPN correction are shown in Figure 3.7 (c) and (d). The target strength required for detection at the specified rates of false alarms and mis-detections are marked by circles in Figures 3.7 (b) and (d). It can be seen that the target strength required when FPN is not corrected ($A_T = 3.22$) is higher than that required when FPN correction is applied ($A_T = 1.86$), implying better performance in the latter case.

3.3.3 Real background an from analog camera

In this case, a real aerial background, obtained from an analog camera used during a flight test was employed. Targets of size 2×2 pixels were synthetically added for mis-detection analysis.

In order to suppress the background, low-stop and morphological pre-processing were separately applied, and the results compared. Since the background was cluttered, a much higher signal was required for satisfactory detection. Even then, the false alarm rate does not reduce sufficiently, thus showing that more post-processing would be required after applying



(a)

(b)

(c)

(d)

Figure 3.7: Results for real noise from camera for 2×2 targets: (a) Plot of MD rate against FA rate (for marked amplitude) for images without FPN correction. The data points are marked as crosses. The MD rate when FA rate is $FA_T = 10$ per simulation is interpolated, and plotted as circle. (b) Plot of MD rate against amplitude for FA rate of $FA_T = 10$ per simulation. The data points are marked as crosses. The value of A_T where MD rate is $MD_T = 0.01$ per target is interpolated and marked as a circle. (c) and (d) Corresponding plots for FPN corrected images.

the algorithm. However, since the number of false alarms (plus true candidates) would be small after this processing, the time complexity of subsequent algorithms would be reduced significantly. The techniques described in Chapter 5 can be used to separate the remaining background clutter from the genuine targets. These techniques utilize the difference in the image translation and expansion between an object on a collision course, and the background clutter.

The false alarm threshold was set $FA_T = 10$ per simulation resulting in a total of 10 false alarms for $N_{FA} = 1$ simulation. The mis-detection threshold was $MD_T = 0.01$ per target, corresponding to 10 mis-detections for $N_{MD} = 20$ simulations with $N_{targ} = 50$ targets. Again, unfortunately, lower rates for false alarm and mis-detection cannot be reliably estimated due to the limited number of background images available.

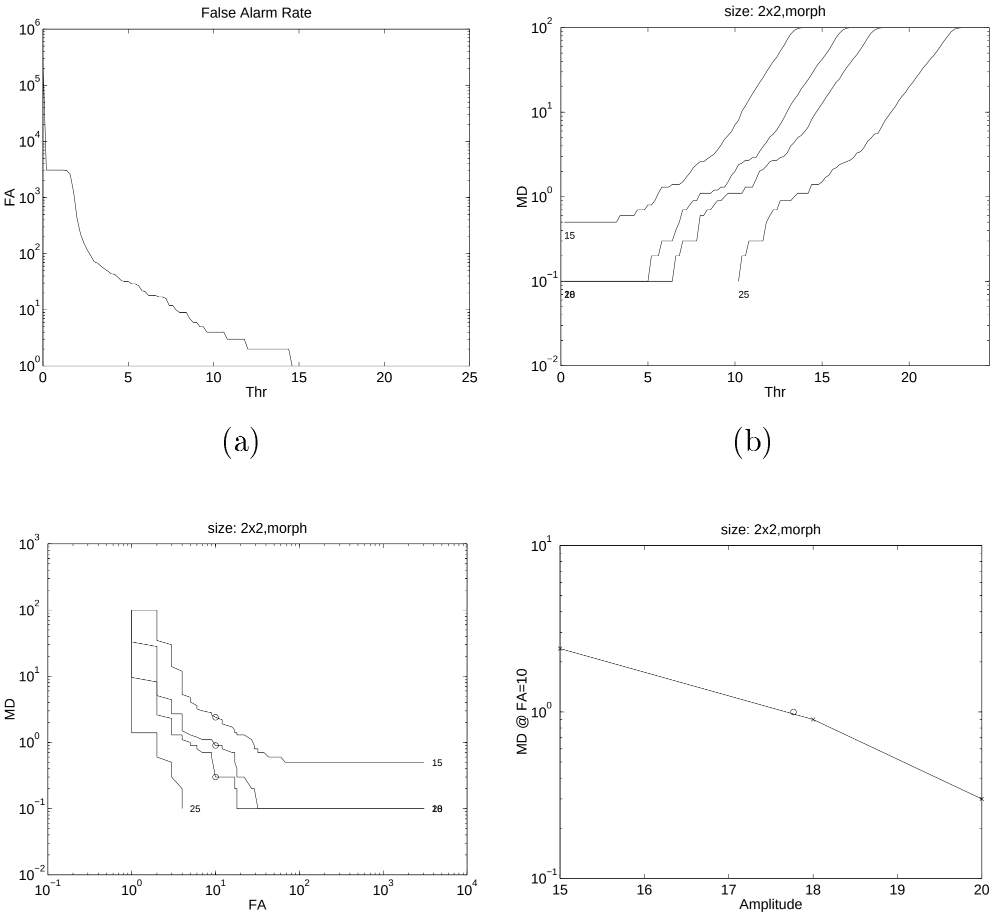
The results for the morphological filter and the low-stop filter are shown in Figures 3.8 and 3.9, respectively. It can be seen that the target strength required when the morphological filter ($A_T = 17.8$) is used is much lower than that required when the low-stop filter ($A_T = 57.8$) is used. The morphological filter is thus better, and the reason for this is that the morphological filter reduces clutter corresponding to large features, whereas the low-stop filter does not do this effectively. However, both result in much poorer performance than that obtained with a digital camera with clear background.

3.3.4 Comparison with other methods

The performance of the dynamic programming algorithm was also compared with other methods such as simple thresholding on a single frame, and temporal averaging on the same number of frames. The comparison was made using FPN correction on images with simulated camera noise. The results of applying the dynamic programming algorithm, simple thresholding on a single frame, and temporal averaging on image sequences with 2×2 moving targets are shown in are shown in Figures 3.10, 3.11 and 3.12 respectively. Temporal averaging was also applied on image sequences with stationary targets instead of moving targets, the results of which are shown in Figure 3.13.

Similar experiments were performed with other target sizes. Table 3.3 shows the comparison the for these algorithms using various target sizes. The plots of the threshold amplitudes against target sizes are shown in Figure 3.14. Again, smaller threshold amplitudes imply better performance as explained before.

It can be seen that the performance of single frame thresholding, as well as temporal



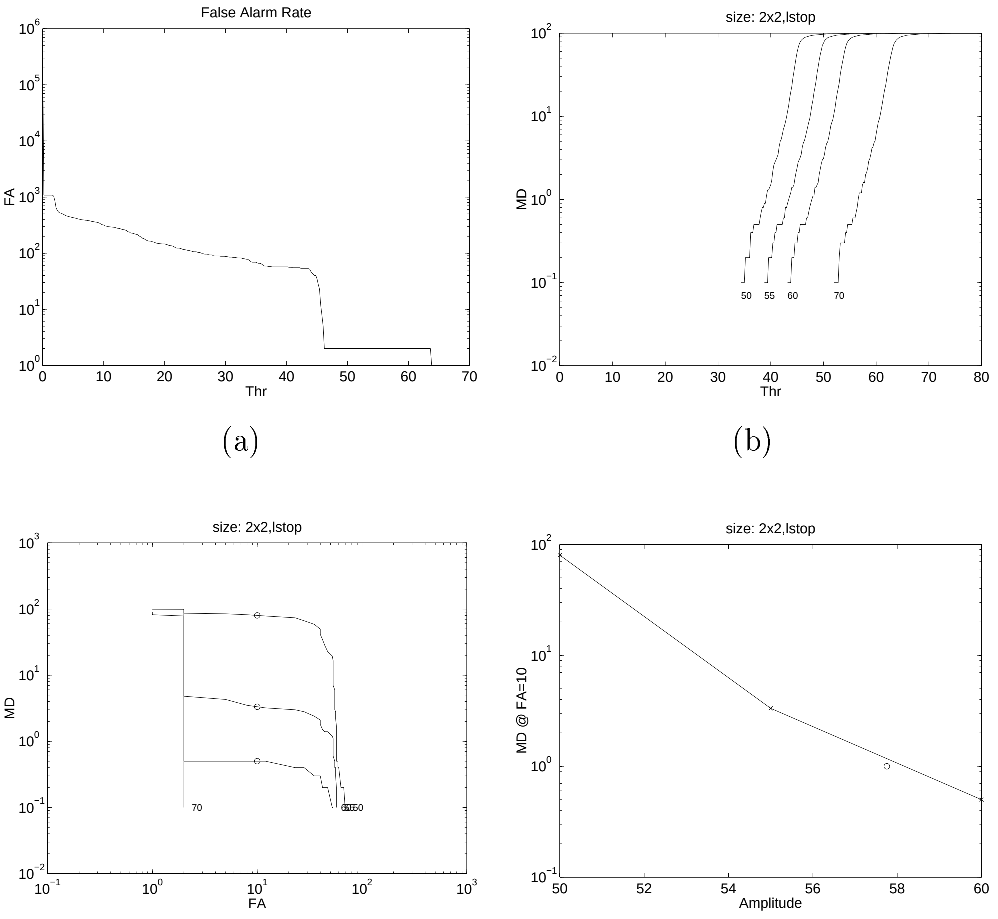
(a)

(b)

(c)

(d)

Figure 3.8: Results for real cluttered background for 2 × 2 targets using morphological filter in the preprocessing: (a) Plot of FA rate (average number per simulation) against threshold (b) Plot of MD rate against threshold, for a number of signal amplitudes (higher amplitudes towards right). (c) Plot of MD rate against FA rate (for marked amplitude). The data points are marked as crosses. The MD rate when FA rate is $FA_T = 10$ per simulation is interpolated, and plotted as circle. (d) Plot of MD rate against amplitude for FA rate of $FA_T = 10$ per simulation. The data points are marked as crosses. The value of A_T where MD rate is $MD_T = 0.01$ per target is interpolated and marked as a circle.



(a)

(b)

(c)

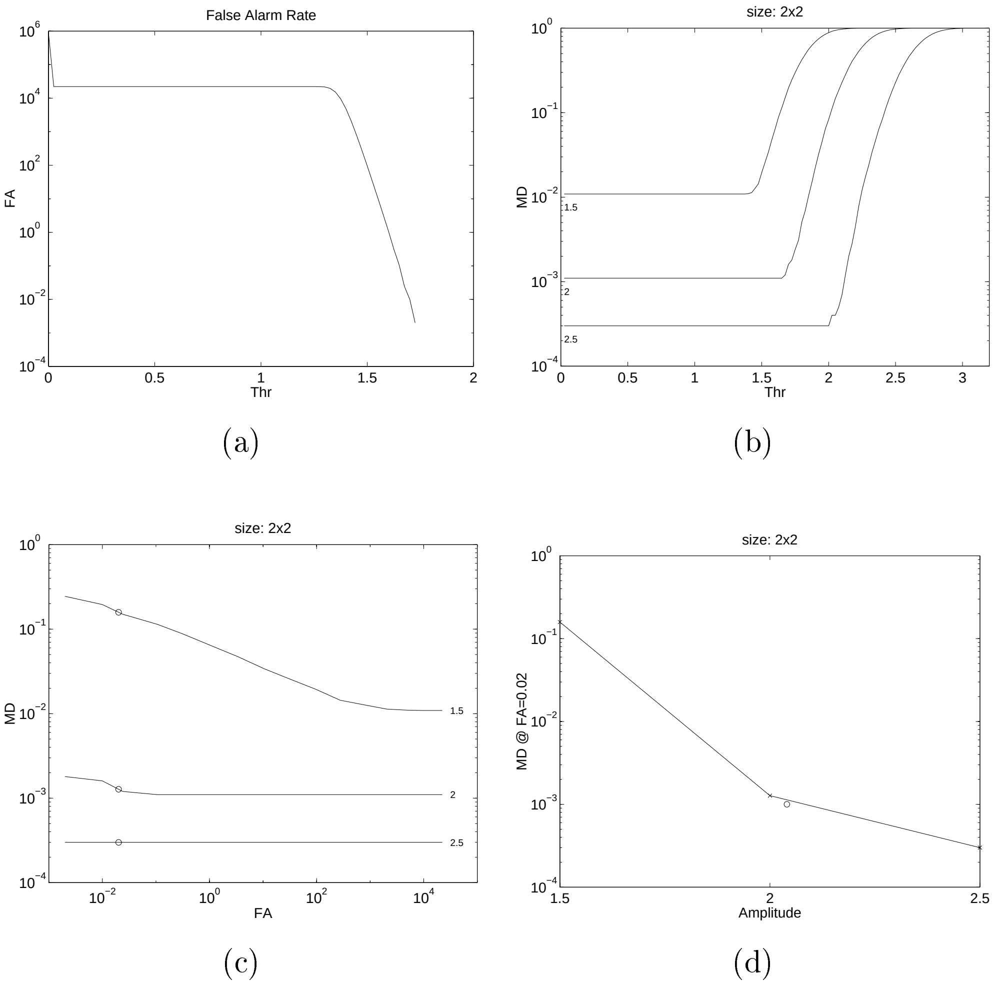
(d)

Figure 3.9: Results for real cluttered background for 2 × 2 targets using low stop filter in the preprocessing: (a) Plot of FA rate (average number per simulation) against threshold (b) Plot of MD rate against threshold, for a number of signal amplitudes (higher amplitudes towards right). (c) Plot of MD rate against FA rate (for marked amplitude). The data points are marked as crosses. The MD rate when FA rate is $FA_T = 10$ per simulation is interpolated, and plotted as circle. (d) Plot of MD rate against amplitude for FA rate of $FA_T = 10$ per simulation. The data points are marked as crosses. The value of A_T where MD rate is $MD_T = 0.01$ per target is interpolated and marked as a circle.

averaging are much poorer than that of the dynamic programming. However, if stationary targets are used instead of moving targets, the performance of temporal averaging is slightly better than that of dynamic programming, showing that temporal averaging is the best choice when the targets are stationary.

Table 3.3: Results of target detection algorithms on simulated image sequences with FPN correction. Threshold amplitudes are shown for false alarm rate of 0.02 per simulation and mis-detection rate of 0.001 per target.

Size	Dynamic prog.	Single frame thresh.	Temp. Avg. (moving)	Temp. Avg. (stat.)
1×1	4.72	23.03	33.82	4.63
1.5×1.5	2.55	10.68	16.99	2.11
2×2	2.04	8.17	11.67	1.65



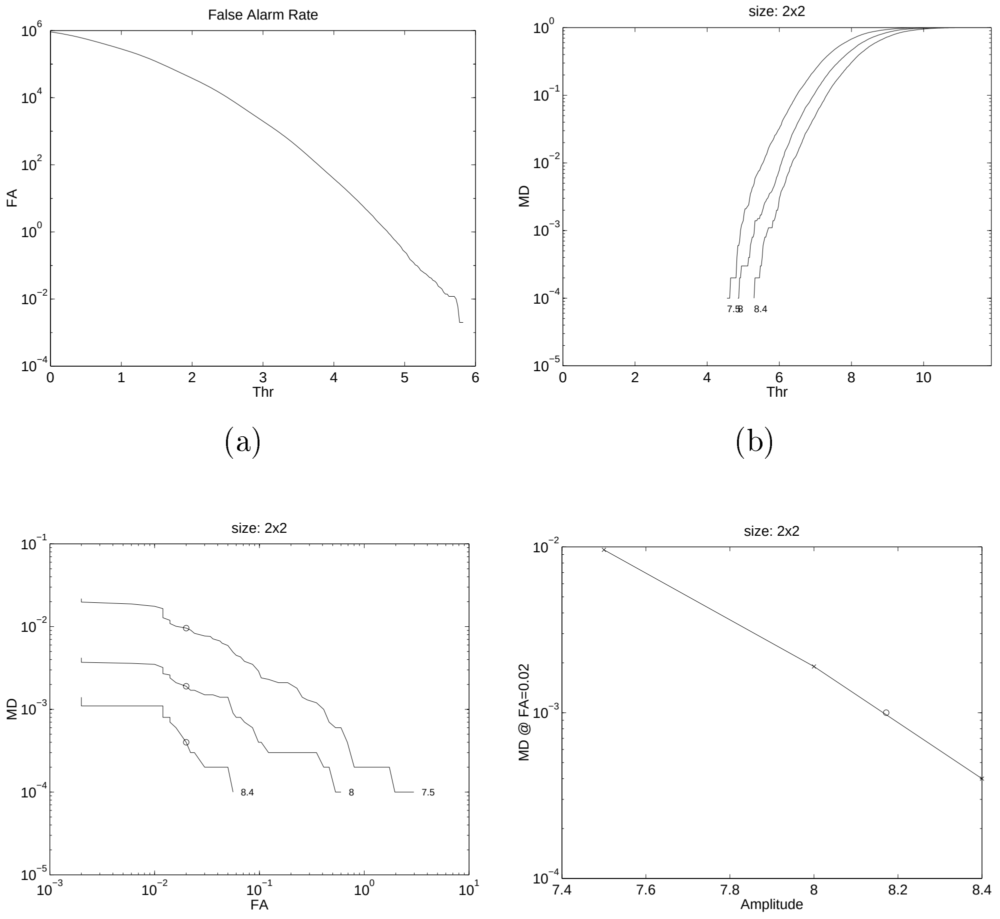
(a)

(b)

(c)

(d)

Figure 3.10: Results for camera noise model with FPN correction for 2×2 targets using dynamic programming: (a) Plot of FA rate (average number per simulation) against threshold (b) Plot of MD rate against threshold, for a number of signal amplitudes (higher amplitudes towards right). (c) Plot of MD rate against FA rate (for marked amplitude). The data points are marked as crosses. The MD rate when FA rate is $FA_T = 0.02$ per simulation is interpolated, and plotted as circle. (d) Plot of MD rate against amplitude for FA rate of $FA_T = 0.02$ per simulation. The value of A_T when MD rate is $MD_T = 0.001$ per target is interpolated and marked as a circle.



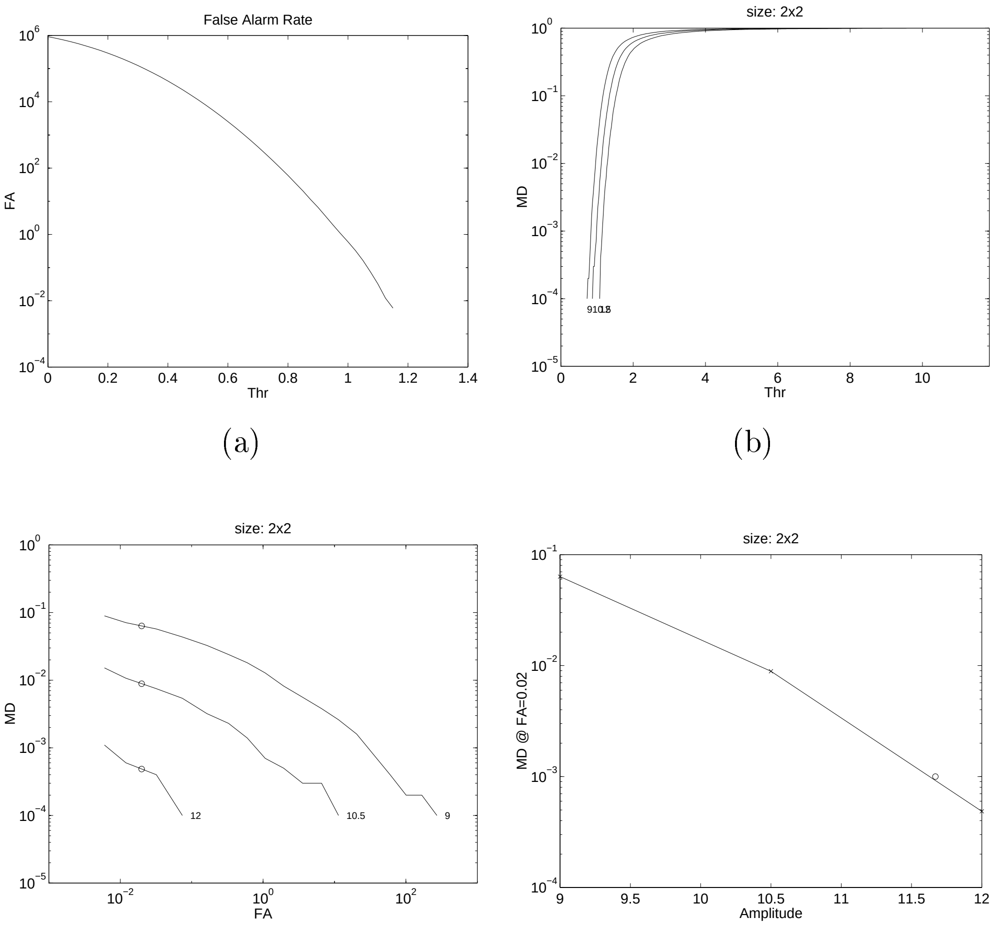
(a)

(b)

(c)

(d)

Figure 3.11: Results for camera noise model with FPN correction for 2 × 2 targets by thresholding a single frame. (a) Plot of FA rate (average number per simulation) against threshold (b) Plot of MD rate against threshold, for a number of signal amplitudes (higher amplitudes towards right). (c) Plot of MD rate against FA rate (for marked amplitude). The data points are marked as crosses. The MD rate when FA rate is $FA_T = 0.02$ per simulation is interpolated, and plotted as circle. (d) Plot of MD rate against amplitude for FA rate of $FA_T = 0.02$ per simulation. The value of A_T when MD rate is $MD_T = 0.001$ per target is interpolated and marked as a circle.



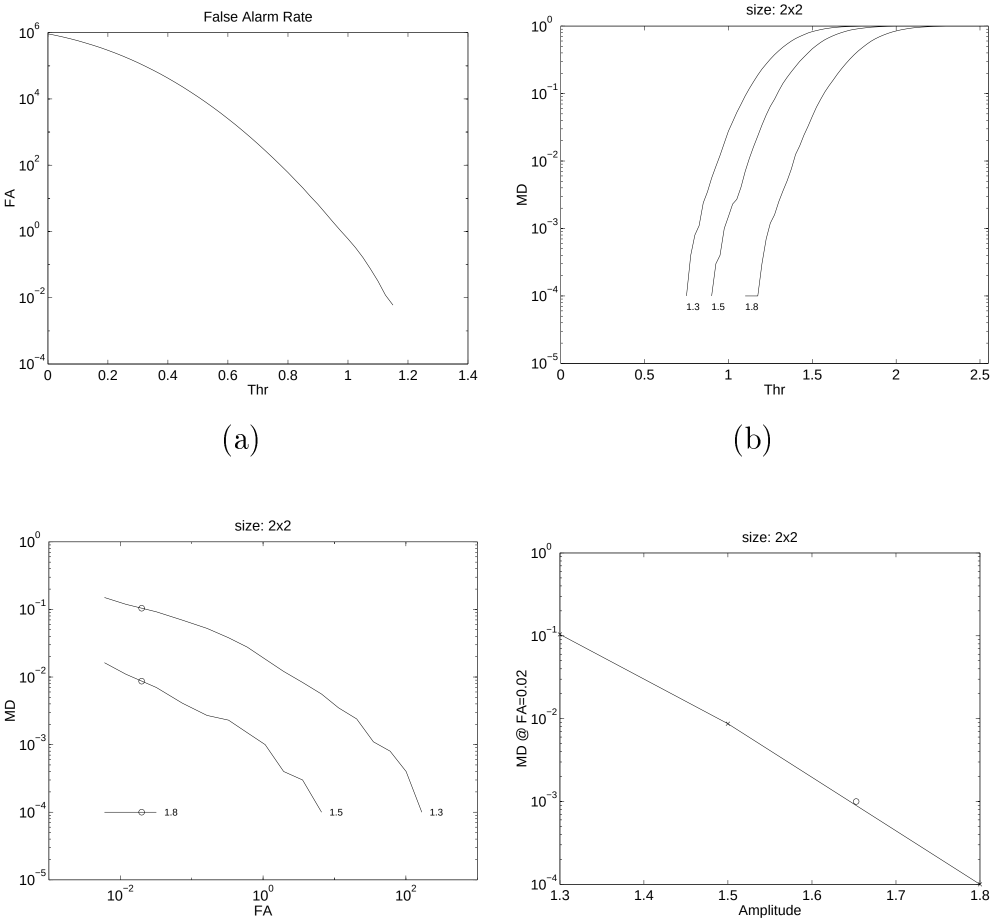
(a)

(b)

(c)

(d)

Figure 3.12: Results for camera noise model with FPN correction for 2 × 2 *moving* targets using temporal averaging. (a) Plot of FA rate (average number per simulation) against threshold (b) Plot of MD rate against threshold, for a number of signal amplitudes (higher amplitudes towards right). (c) Plot of MD rate against FA rate (for marked amplitude). The data points are marked as crosses. The MD rate when FA rate is $FA_T = 0.02$ per simulation is interpolated, and plotted as circle. (d) Plot of MD rate against amplitude for FA rate of $FA_T = 0.02$ per simulation. The value of A_T when MD rate is $MD_T = 0.001$ per target is interpolated and marked as a circle.



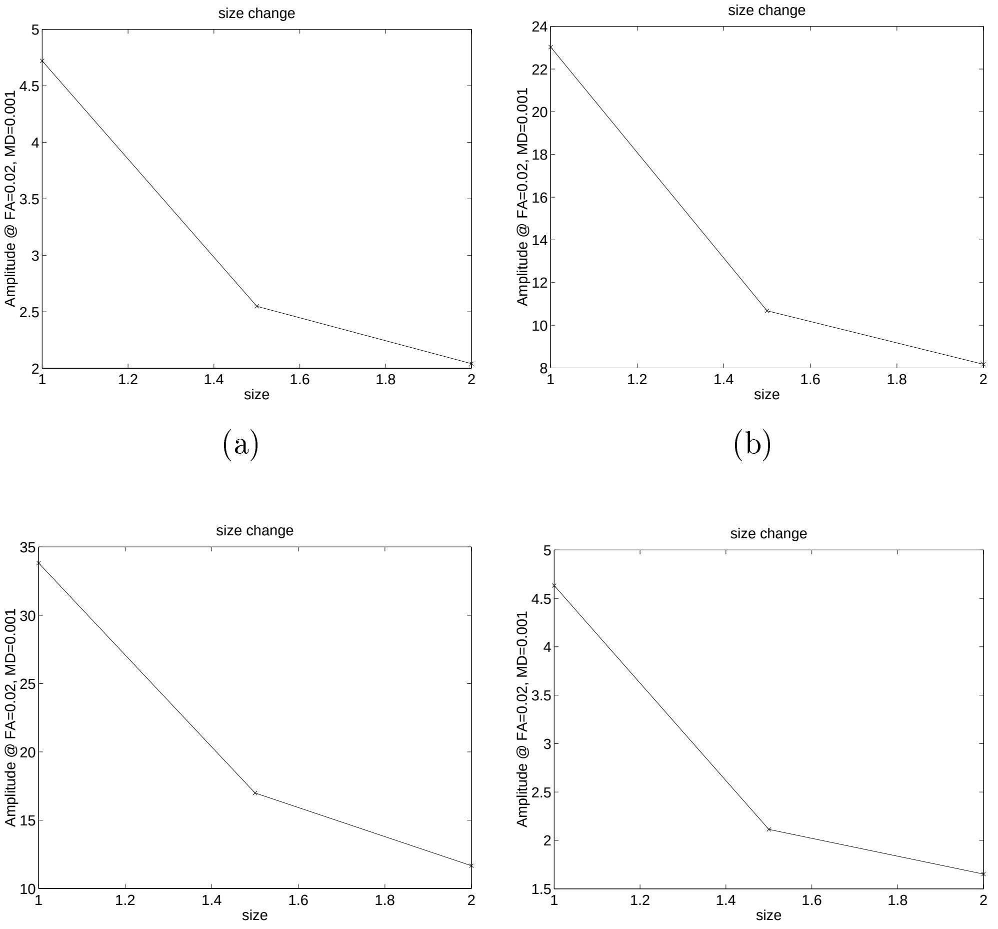
(a)

(b)

(c)

(d)

Figure 3.13: Results for camera noise model with FPN correction for 2×2 *stationary* targets using temporal averaging. (a) Plot of FA rate (average number per simulation) against threshold (b) Plot of MD rate against threshold, for a number of signal amplitudes (higher amplitudes towards right). (c) Plot of MD rate against FA rate (for marked amplitude). The data points are marked as crosses. The MD rate when FA rate is $FA_T = 0.02$ per simulation is interpolated, and plotted as circle. (d) Plot of MD rate against amplitude for FA rate of $FA_T = 0.02$ per simulation. The value of A_T when MD rate is $MD_T = 0.001$ per target is interpolated and marked as a circle.



(a)

(b)

(c)

(d)

Figure 3.14: Performance comparison of several algorithms: Plot of amplitude against the target size ($x \times x$) for experiments without FPN correction using (a) Dynamic programming, (b) Thresholding single frame, (c) Temporal averaging (moving targets), (d) Temporal averaging (stationary targets).

Chapter 4

Theoretical Performance of Detection Algorithms

In this chapter, the approximate theoretical performance of the algorithms presented in Chapter 2 are derived. The theoretical derivations are based on the paper by Tonissen and Evans [18]. The theoretical performance is compared with the experimentally observed performance described in Chapter 3. Effects of approximations used in the derivations are also described.

4.1 Dynamic programming algorithm

The dynamic programming algorithm described in Chapter 2 can be summarized as follows:

1. Initialization: For all pixels (x, y) and all velocities (u, v), set

$$F(x, y; u, v; 0) = 0$$

2. Recursion: At time k, set

$$F(x, y; u, v; k) = (1 - \alpha) f(x, y; k) + \alpha \max_{(x', y') \in Q} F(x - u - x', y - v - y'; u, v; k - 1)$$

3. Termination: At time K, take

$$F_{max}(x, y; K) = \max_{(u,v) \in P} F(x, y; u, v; K)$$

The number of elements in sets P and Q are denoted by p and q, respectively. The values of $p = q = 4$ have been used in our implementation, with $u, v \in \{-1, 0\}$ and $x', y' \in \{0, 1\}$. Note that for theoretical analysis, the recursion step is replaced by:

$$F(x, y; u, v; k) = f(x, y; k) + \alpha \max_{(x',y') \in Q} F(x - u - x', y - v - y'; u, v; k - 1) \tag{4.1}$$

However, this only changes F by a scale factor, and since both signal as well as noise would be scaled equally, SNR analysis does not change.

4.2 False alarm and mis-detection probabilities

Probability of false alarms P_{FA} is the probability that there is at least one state exceeding the threshold V_T out of p velocity states at the final output time K, for the pixel where there is no signal in its neighborhood – i.e., hypothesis H_0.

$$P_{FA}(x, y) = Pr\left[\max_{(u,v) \in P} F(x, y; u, v; K) \geq V_T | H_0\right] = 1 - [P_{0,K}(V_T)]^p \tag{4.2}$$

where $P_{0,K}(V_T)$ denotes the probability of F for hypothesis H_0 at time K, being less than or equal to the threshold V_T.

Probability of mis-detection P_{MD} is the probability that there is no output with correct velocity (u, v) exceeding the threshold at time K, within a neighborhood R of size $r + 1$, where one cell contains signal – i.e., hypothesis H_1 – and the other r cells are noise. This allows for some tolerance in the location of target. For example, a 5 × 5 neighborhood corresponding to $r + 1 = 25$ gives a tolerance of ±2 pixels in the location of the target. On the other hand, a 1 × 1 neighborhood consisting of only the target position corresponds to $r + 1 = 1$ or $r = 0$ giving no tolerance for the target position.

$$\begin{aligned} P_{MD}(x, y; u, v) &= Pr\left[\max_{x',y' \in R} F(x - x', y - y'; u, v; K) \leq V_T | H_1\right] \\ &= P_{1,K}(V_T)[P_{0,K}(V_T)]^r \end{aligned} \tag{4.3}$$

where $P_{1,K}(V_T)$ denotes the probability of F for hypothesis H_1 at time K being less than or equal to the threshold V_T.

4.3 Normal approximations

For an analytic solution of the performance of the dynamic programming algorithm, the distributions of the intermediate outputs can be approximated using normal approximations.

Table 4.1: Values of μ_q and σ_q^2 for a number of values of q.

q	μ_q	σ_q^2
1	0	1
4	1.029	0.491
9	1.485	0.3574
25	1.965	0.2585
49	2.241	0.2168

Consider q independent standard normal variables $w_i \sim N(0,1)$. The cumulative distribution function (CDF) of the maximum of these variables is given by:

$$P(w) = Pr\left[\max_i w_i \leq w\right] = \prod_i Pr[w_i \leq w] = [\Phi(w)]^q \tag{4.4}$$

where $\Phi(\cdot)$ is the CDF of a standard normal variable. The probability density function (PDF) is the derivative of the CDF given by:

$$p(w) = q[\Phi(w)]^{q-1}G(w) \tag{4.5}$$

where $G(\cdot)$ is the standard normal PDF.

This distribution of maximum of q standard normal variables can be approximated as a normal distribution $N(\mu_q, \sigma_q^2)$, where μ_q and σ_q^2 denote the mean and the variance of the actual distribution. These are computed using numerical integration, and are tabulated in Table 4.1 for different values of q.

For general normal variables $z_i \sim N(\mu, \sigma^2)$, one can substitute: $z_i = \mu + \sigma w_i$ where w_i are standard normal variables. The maximum of z_i is approximately normally distributed with mean and variance given by:

$$\begin{aligned} E[\max z_i] &= \mu + \sigma E[\max w_i] = \mu + \sigma\mu_q \\ V[\max z_i] &= \sigma^2 V[\max w_i] = \sigma^2\sigma_q^2 \end{aligned} \tag{4.6}$$

Let the input at any time k be normally distributed, both in absence and presence of the target, so that:

$$f(x,y;k|H_0) \sim N(\mu_n, \sigma_n^2)\ ,\ f(x,y;k|H_1) \sim N(\mu_s, \sigma_s^2) \tag{4.7}$$

Then, the distributions of the output F at time k will also be *approximately* normally distributed so that

$$F(x,y;u,v;k|H_0) \simeq N(M_{0,k}, S_{0,k}^2)\ ,\ F(x,y;u,v;k|H_1) \simeq N(M_{1,k}, S_{1,k}^2) \tag{4.8}$$

where the M and S parameters are calculated below.

4.4 False alarm analysis

For noise pixels, we have:

$$\begin{aligned} F(x,y;u,v;0) &= 0 \\ F(x,y;u,v;k) &= f(x,y;k) + \alpha \max_{(x',y')\in Q} F(x-u-x', y-v-y'; u,v;k-1) \\ &\simeq N(M_{0,k}, S_{0,k}^2) \end{aligned} \tag{4.9}$$

Using equation (4.6), the mean and variance parameters at time k can be recursively expressed as:

$$\begin{aligned} M_{0,0} = 0\ ,\ M_{0,k} &= \mu_n + \alpha(M_{0,k-1} + \mu_q S_{0,k-1}) \\ S_{0,0}^2 = 0\ ,\ S_{0,k}^2 &= \sigma_n^2 + \alpha^2\sigma_q^2 S_{0,k-1}^2 \end{aligned} \tag{4.10}$$

Solving these recursive equations yields expressions for mean and variance at time K:

$$\begin{aligned} S_{0,K}^2 &= \sigma_n^2 \frac{1-\alpha^{2K}\sigma_q^{2K}}{1-\alpha^2\sigma_q^2} \\ M_{0,K} &= \frac{1-\alpha^K}{1-\alpha}\mu_n + \alpha\mu_q \sum_{i=0}^{K-1} (\alpha^i S_{0,K-i-1}) \end{aligned} \tag{4.11}$$

To get approximate closed-form expressions for $M_{0,K}$, one can write $S_{0,k}$ as:

$$S_{0,k} = \sigma_n \sqrt{\frac{1-\alpha^{2K}\sigma_q^{2K}}{1-\alpha^2\sigma_q^2}} \simeq \frac{\sigma_n}{\sqrt{1-\alpha^2\sigma_q^2}} \left(1 - \gamma_k \alpha^{2k}\sigma_q^{2k}\right) \tag{4.12}$$

where γ_k is dependent on k but always lies between 0 and 1. Using $\gamma_k = 1/2$ is equivalent to using the first order term of binomial expansion, whereas $\gamma_k = 0$ corresponds to assuming that $S_{0,k}$ remains approximately constant with k, which is justifiable, since σ_q^2 is quite small. Accordingly, we have:

$$\begin{aligned} M_{0,K} &= \frac{1-\alpha^K}{1-\alpha}\mu_n + \alpha\mu_q \sum_{k=0}^{K-1} (\alpha^k S_{0,K-k-1}) \\ &= \frac{1-\alpha^K}{1-\alpha}\mu_n + \frac{\alpha\mu_q\sigma_n}{\sqrt{1-\alpha^2\sigma_q^2}} \left[\frac{1-\alpha^K}{1-\alpha} - \gamma\alpha^{K-1}\frac{1-(\alpha\sigma_q^2)^K}{1-\alpha\sigma_q^2}\right] \end{aligned} \tag{4.13}$$

where γ is a function of all γ_k and also lies between 0 and 1. Values of $\gamma = 0$ and $\gamma = 1/2$ can be used as the zero order and first order approximations, respectively. For $K \to \infty, \alpha \neq 1$ such that $\alpha^K \ll 1$ (also, $\sigma_q^{2K} \ll 1$), we have:

$$\begin{aligned} S_{0,K}^2 &= \frac{\sigma_n^2}{1-\alpha^2\sigma_q^2} \\ M_{0,K} &= \frac{1}{1-\alpha}\left[\mu_n + \frac{\alpha\mu_q\sigma_n}{\sqrt{1-\alpha^2\sigma_q^2}}\right] \end{aligned} \tag{4.14}$$

For the case when $\alpha = 1$, the sum $\sum_{i=0}^{K} \alpha^i$ changes from $(1-\alpha^K)/(1-\alpha)$ to K. Hence, the expressions become:

$$\begin{aligned} S_{0,K}^2 &= \sigma_n^2 \frac{1-\sigma_q^{2K}}{1-\sigma_q^2} \\ M_{0,K} &= K\mu_n + \frac{\mu_q\sigma_n}{\sqrt{1-\sigma_q^2}}\left[K - \gamma\frac{1-(\sigma_q^2)^K}{1-\sigma_q^2}\right] \end{aligned} \tag{4.15}$$

Finally, the probability of false alarms is:

$$P_{FA} = 1 - [P_{0,K}(V_T)]^p \tag{4.16}$$

giving

$$P_{0,K}(V_T) = (1-P_{FA})^{1/p} = \Phi\left(\frac{V_T - M_{0,K}}{S_{0,K}}\right) \tag{4.17}$$

where $\Phi(\cdot)$ denotes the CDF of a standard normal variable. Hence, the threshold V_T can be expressed in terms of the mean $M_{0,K}$, variance $S_{0,K}$, and the false alarm probability P_{FA} as:

$$V_T = M_{0,K} + S_{0,K}\Phi^{-1}[(1-P_{FA})^{1/p}] = M_{0,K} + S_{0,K}\phi_{0,p} \tag{4.18}$$

where

$$\phi_{0,p} = \Phi^{-1}[(1-P_{FA})^{1/p}] \simeq \Phi^{-1}[1-P_{FA}/p] \tag{4.19}$$

4.5 Missed detection analysis

The probability of mis-detection is given by:

$$P_{MD} = P_{1,K}(V_T)[P_{0,K}(V_T)]^r \leq P_{1,K}(V_T) \tag{4.20}$$

Substituting the expression of V_T in terms of false alarm rate, we have:

$$P_{MD} = (1 - P_{FA})^{r/p} P_{1,K}(V_T) \tag{4.21}$$

giving

$$P_{1,K}(V_T) = \frac{P_{MD}}{(1 - P_{FA})^{r/p}} = \Phi\left(\frac{V_T - M_{1,K}}{S_{1,K}}\right) \tag{4.22}$$

Hence,

$$V_T = M_{1,K} + S_{1,K}\Phi^{-1}\left[\frac{P_{MD}}{(1 - P_{FA})^{r/p}}\right] = M_{1,K} - S_{1,K}\,\phi_{1,p} \tag{4.23}$$

where

$$\phi_{1,p} = -\Phi^{-1}\left[\frac{P_{MD}}{(1 - P_{FA})^{r/p}}\right] = \Phi^{-1}\left[1 - \frac{P_{MD}}{(1 - P_{FA})^{r/p}}\right] \simeq \Phi^{-1}\left[1 - P_{MD}\right] \tag{4.24}$$

since usually, $P_{FA} \ll 1$.

Approximations of $M_{1,K}$ and $S^2_{1,K}$, are obtained considering the exceeding of threshold only due to the signal part, and not due to the noise part. Also, it is assumed that the target occupies a single pixel. In such a case, we have:

$$F(x, y; u, v; k) \simeq f(x, y; k) + \alpha F(x, y; u, v; k - 1) \simeq N(M_{1,k}, S^2_{1,k}) \tag{4.25}$$

It can be easily shown that:

$$M_{1,K} \simeq \frac{1 - \alpha^K}{1 - \alpha}\mu_s\ ,\ S^2_{1,K} \simeq \frac{1 - \alpha^{2K}}{1 - \alpha^2}\sigma^2_s \tag{4.26}$$

4.6 Calculation of required SNR

To calculate the SNR required for detection at particular rates of false alarms and mis-detections, equations (4.18) and (4.23) are combined to give:

$$M_{1,K} - M_{0,K} = S_{0,K}\,\phi_{0,p} + S_{1,K}\,\phi_{1,p} \tag{4.27}$$

Using expressions for $S_{0,K}$, $M_{0,K}$, $S_{1,K}$, and $M_{1,K}$, and assuming $\mu_n = 0$, $\mu_s = \mu$, and $\sigma_n = \sigma_s = \sigma$, equation (4.27) becomes:

$$\begin{aligned} \frac{1-\alpha^K}{1-\alpha}\mu - \frac{\sigma\alpha\mu_q}{\sqrt{1-\alpha^2\sigma_q^2}}\left[\frac{1-\alpha^K}{1-\alpha} - \gamma\alpha^{K-1}\frac{1-\alpha^K\sigma_q^{2K}}{1-\alpha\sigma_q^2}\right] & \\ \simeq \sigma\sqrt{\frac{1-\alpha^{2K}\sigma_q^{2K}}{1-\alpha^2\sigma_q^2}}\,\phi_{0,p} + \sigma\sqrt{\frac{1-\alpha^{2K}}{1-\alpha^2}}\,\phi_{1,p} & \qquad (4.28) \end{aligned}$$

The SNR required for detection is given by:

$$\begin{aligned} SNR_T &= \frac{\mu}{\sigma} \simeq \frac{\alpha\mu_q}{\sqrt{1-\alpha^2\sigma_q^2}}\left[1 - \gamma\alpha^{K-1}\frac{1-\alpha}{1-\alpha^K}\cdot\frac{1-\alpha^K\sigma_q^{2K}}{1-\alpha\sigma_q^2}\right] \\ &+ \frac{1-\alpha}{1-\alpha^K}\sqrt{\frac{1-\alpha^{2K}\sigma_q^{2K}}{1-\alpha^2\sigma_q^2}}\,\phi_{0,p} + \sqrt{\frac{1-\alpha}{1+\alpha}\cdot\frac{1+\alpha^K}{1-\alpha^K}}\,\phi_{1,p} \qquad (4.29) \end{aligned}$$

For $\alpha = 1$, replacing $(1-\alpha^K)/(1-\alpha)$ by K, we get

$$SNR_T = \frac{\mu}{\sigma} \simeq \frac{\mu_q}{\sqrt{1-\sigma_q^2}}\left[1 - \frac{\gamma}{K}\cdot\frac{1-\sigma_q^{2K}}{1-\sigma_q^2}\right] + \frac{1}{K}\sqrt{\frac{1-\sigma_q^{2K}}{1-\sigma_q^2}}\,\phi_{0,p} + \frac{1}{\sqrt{K}}\,\phi_{1,p} \qquad (4.30)$$

For $K \to \infty, \alpha \neq 1$ such that $\alpha^K \ll 1$:

$$SNR_T = \frac{\mu}{\sigma} \simeq \frac{\alpha\mu_q}{1-\alpha^2\sigma_q^2} + \frac{1-\alpha}{\sqrt{1-\alpha^2\sigma_q^2}}\,\phi_{0,p} + + \sqrt{\frac{1-\alpha}{1+\alpha}}\,\phi_{1,p} \qquad (4.31)$$

The above expressions of SNR_T can be written in the form:

$$SNR_T = A + B\,\phi_{0,p} + C\,\phi 1, p \qquad (4.32)$$

where A, B, and C depend on K, q, and α. The terms B and C decrease with K, improving the algorithm performance as K increases. However, the term A *increases* with K, putting a lower bound on the *required* SNR, thus limiting the performance. It can be shown that this bound increases with q, and hence a lowest possible value of q should be used. This is intuitively explained, since a maximum is taken over q noise pixels and it is more likely to be a false alarm when q is large.

4.7 Temporal averaging and single frame thresholding as special cases

Recursive temporal averaging algorithm can be considered as a special case of dynamic programming with $p = q = 1$, for which $\mu_q = 0$ and $\sigma_q^2 = 1$. Hence, the threshold SNR for recursive temporal averaging becomes:

$$SNR_T = \frac{\mu}{\sigma} \simeq \sqrt{\frac{1-\alpha}{1+\alpha} \cdot \frac{1+\alpha^K}{1-\alpha^K}} \left[\phi_{0,1} + \phi_{1,1}\right] \tag{4.33}$$

This expression can also be obtained by using the recursive temporal averaging equations:

$$F(x, y; 0) = 0 \ , \ F(x, y; k) = f(x, y; k) + \alpha F(x, y; k-1) \tag{4.34}$$

Also, for $\alpha = 1$, this expression takes the limit:

$$SNR_T = \frac{1}{\sqrt{K}} \left[\phi_{0,1} + \phi_{1,1}\right] \tag{4.35}$$

The same result would be obtained by using $\alpha = 1$ in original equations. For $K \to \infty, \alpha \neq 1$ such that $\alpha^K \ll 1$,

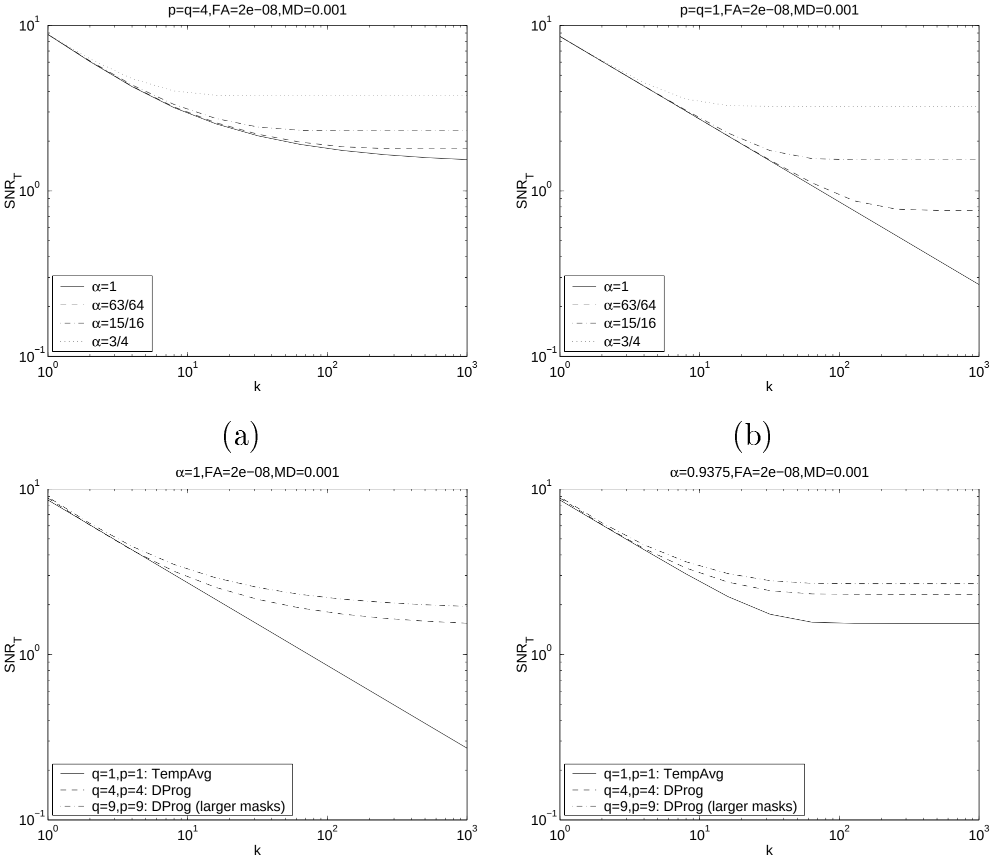
$$SNR_T = \sqrt{\frac{1-\alpha}{1+\alpha}} \left[\phi_{0,1} + \phi_{1,1}\right] \tag{4.36}$$

For single frame thresholding ($K = 1$ or $\alpha = 0$), the threshold SNR reduces to $\phi_{0,1} + \phi_{1,1}$.

Note that the first term from the dynamic programming algorithm disappears in these expressions, and there is no lower limit to the performance if $\alpha = 1$.

4.8 Theoretical performance plots

This section describes the behavior of the required signal to noise ratio SNR_T for different values of parameters. It should be noted that lower *required* SNR means *better* performance. Figure 4.1 (a) shows plots of SNR_T against K for dynamic programming algorithm with $p = q = 4$ and a number of values of α. The false alarm rate is 2×10^{-8} (0.02 per simulation for a 1 mega-pixel image), and the mis-detection rate is 0.001. It can be seen that SNR_T decreases with increase in K, but saturates at a certain point depending on α. Figure 4.1 (b) shows the corresponding plot for $p = q = 1$ – i.e., recursive temporal averaging. Figures 4.1 (c) and (d) show the plots of SNR_T against K with $\alpha = 1$ and $\alpha = 15/16$, respectively, for a number of values of p and q. It is observed that SNR_T increases with q as expected. The SNR_T also increases slightly with p, but the plots cannot show the change. Except in the case of $\alpha = 1$ and $p = q = 1$ – i.e., temporal averaging – the SNR_T saturates at some minimum value as $K \to \infty$.



(a) (b)

(c) (d)

Figure 4.1: Plots of SNR_T against K for: (a) $p = q = 4$ (dynamic programming) and number of α values. (b) $p = q = 1$ (temporal averaging) and number of α values. (c) $\alpha = 1$ and number of p and q values. (d) $\alpha = 15/16$ and number of p and q values. The parameters used are: $FA = 2 \times 10^{-8}, MD = 0.001$.

Table 4.2: Parameters used for calculating the theoretical performance of algorithms.

Parameter	Dynamic prog	Single frame	Temp. Avg. (stat)
FA	$2 \times 10^{-8}/pixel = 0.02/image$		
MD	$0.001/pixel$		
α	15/16		
K	32	1	32
q	4	—	1

Table 4.3: Comparison of theoretical performance of the algorithms with observed performance on 2×2 targets.

Algorithm	Theoretical SNR	Observed SNR
Dynamic Prog	2.4540	2.04
Single frame	8.5811	8.17
Temp. Avg. (stat.)	1.7507	1.65

4.9 Comparison between theoretical and observed performance

The parameters used in the calculation of theoretical performance of the algorithms for 2×2 targets are shown in Table 4.2. The calculated and the observed SNR threshold for these parameters for various algorithms are shown in Table 4.3.

One can observe that the actual performance of the algorithm for 2×2 targets is slightly better than the theoretical performance for most of the algorithms. The reason for this is, that a 2×2 target occupies at least one pixel completely, and a few other pixels partially. Hence, its performance should be slightly greater than the calculated performance in which one assumes that the target occupies exactly one pixel.

To correct this problem, point targets were used in place of 2×2 targets. The experiments in Chapter 3 were repeated using point targets. The comparison between the calculated and observed SNR for a number of false alarm and mis-detection rates are shown in Table 4.4.

Table 4.4: Comparison of theoretical performance of the algorithms with observed performance on point targets for a number of different values of false alarm (FA) and mis-detection (MD) rates.

Algorithm	FA rate	MD rate	Theo. SNR	Obs. SNR
Dynamic Prog.	2×10^{-8}=0.02/simul	0.001	2.4540	2.7172
Dynamic Prog.	10^{-6}=1/simul	0.01	2.2313	2.2928
Dynamic Prog.	10^{-6}=1/simul	0.1	2.0181	1.9862
Dynamic Prog.	10^{-4}=100/simul	0.1	1.9259	1.8401
Temp. Avg.	2×10^{-8}=0.02/simul	0.001	1.7507	1.7355
Temp Avg.	10^{-6}=1/simul	0.01	1.4444	1.4307
Temp Avg.	10^{-6}=1/simul	0.1	1.2313	1.2345

It can be seen that the calculated and observed SNR rates agree very well in most cases. However, in the case of extremely low false alarm and mis-detection rates, the observed SNR is greater than the calculated SNR for the dynamic programming algorithm. The reason for this is the normal approximation used for the distribution of resulting output.

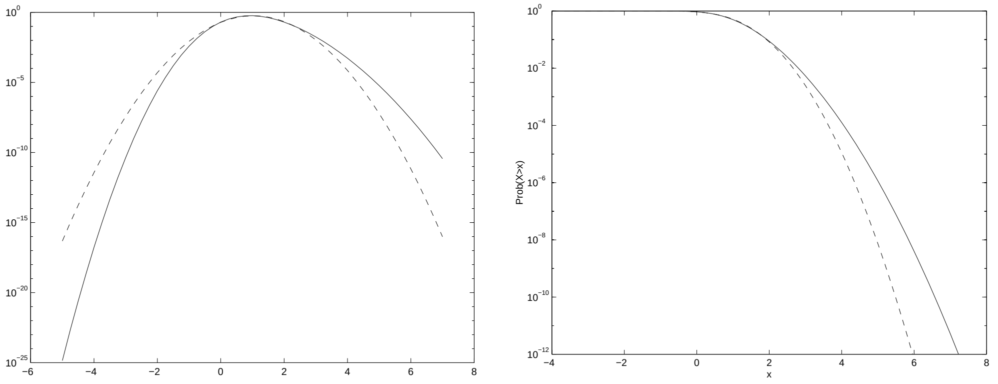
4.10 Effect of approximations

Approximations were used to derive the closed form expressions. In this section, the effects of these approximations are described.

Normal approximation

Normal approximation was used for maximum of q normal variables. The comparison of the probability density, and the complementary cumulative distribution functions of the maximum of $q = 4$ standard normal variables, and their normal approximation are shown in Figure 4.2. It can be seen that the approximation is good in the interior, where probability density is high, but is inaccurate in the tails, where the probability density is low.

Due to the difference in these distributions, the probability of false alarms is underestimated. In fact, to get the actual value of the false alarm rate, the function corresponding to the actual cumulative distribution of the output F should be used in place of cumulative



(a) (b)

Figure 4.2: Probability distributions of normal approximations: (a) Probability density function ($p(x)$ and $p_n(x)$) (b) Complementary cumulative distribution ($Q(x)$ and $Q_n(x)$) against x of the maximum of $q = 4$ standard normal variables (solid line) and the normal approximation having same mean and variance.

normal distribution. But this distribution is difficult to obtain in closed form.

To get an idea of the difference between the actual distribution and the normal approximation, consider the function corresponding to the complementary cumulative distribution $Q(x) = Pr[X > x]$ of the maximum of $q = 4$ normal variables as shown in Figure 4.2. For $Q(x) = 10^{-8}$, we get $x = 5.85$, whereas for normal distribution the corresponding $Q_n(x) = 10^{-8}$ gives $x = 4.95$. The difference is around 18 % but is smaller for smaller values of x.

At each step of the recursion, maximum of q instances of F at time $k-1$ are taken and added to the input f at time k to obtain the output F at time k. Hence, the distribution of the output F at each time should be a better approximation of normal distribution than $Q(\cdot)$, since a normal variable (f) is added to the maximum term for obtaining the output F. Also, since the normal approximation for F is good in the interior, the mean and variance of maximum of q instances of F will be close to what is computed assuming the normal distribution. Hence, the mean and variance calculations are not affected much.

Furthermore, it is observed that the threshold SNR changes are small even for large changes in false alarm and mis-detection rates. In any case, one would not directly use the false alarm and mis-detection rates during the application of the algorithm, but estimate these dynamically using the output from the algorithm.

Approximation in false alarm estimation

Another approximation was performed while computing the mean value $M_{0,K}$ of the noise output, used in false alarm estimation. For equation (4.12), γ actually depends on k, which makes it impossible to get an exact analytical expression. It was assumed that γ is fixed and approximately equal to 1/2, corresponding to a first order approximation. However, it is observed that the value of $M_{0,K}$ does not change much with γ even for the extremes of $\gamma = 0$ or $\gamma = 1$. Hence, the approximation is reliable.

Approximation in mis-detection estimation

In the case of mis-detections, the output of the algorithm at a target point is assumed to be solely due to the target, without the effect of noise. The noise can add or subtract the target intensity. However, since maximum is taken over q pixels at every stage, bias is likely towards adding. Hence, the mis-detections are likely to be less than what are estimated.

Chapter 5

A Special Approach for Hazard Detection

It is well known in the pilots' community, that an object on a collision or near-collision course remains stationary or nearly stationary in its 2-D image view [14]. The closest distance that an aircraft would approach another before moving away from it, is known as the distance of passage, and the time to reach that point is known as the time to passage, or time to 'collision'. For ensuring safety, the distance of passage should be larger than a certain limit; and objects with a smaller distance of passage should be detected before the time to collision becomes too small. It can be shown that the rate of translation of the object in the image is proportional to the distance of passage. Using this property, the rate of image translation can be used to separate hazardous objects from clutter, since the former have a smaller rate of translation.

Another useful property which can be used to discriminate hazardous objects from clutter is the rate of image expansion, which is approximately inversely proportional to the time to collision of the object. Nelson and Aloimonos [15] use the image expansion in terms of the flow field divergence to estimate the time to collision, for separating obstacles. Francois and Bouthemy [7] separate the image motion into components of divergence, rotation, and deformation. Ancona and Poggio [1] use 1-D correlation to estimate optical flow for a time-to-crash detector. Baram and Barniv [3] rely on object texture to extract information on local expansion. Instead of estimating a numerical depth value, an object is classified as 'safe' or 'dangerous' using a pattern recognition approach.

Most of these methods are useful for objects of larger sizes. However, in this case, the object sizes can be very small, even sub-pixel, along with very small rates of expansion.

Hence, a feature based approach was used in this work, where features were tracked, and their expansion estimated over a large number of frames.

This chapter describes the conditions under which the rates of image translation and expansion can be used to separate an object on collision course from the ground clutter. Methods to estimate the image translation and expansion are proposed and tested on real image sequences obtained from a camera mounted on an aircraft.

5.1 Scene geometry

Consider an object approaching towards the aircraft with a *relative* velocity of V as shown in Figure 5.1 (a). Let p be the distance of passage – i.e., the closest distance that the object approaches the camera – and ϕ be the angle between the line of sight of the target and the relative velocity vector V. Let τ denote the time to passage (or collision) which is the time the object takes to reach the distance of passage. The object distance is r, whereas distance that the object travels until it reaches the point of passage P is z.

5.2 Detection using translation

As the object moves, the angle ϕ as well as distances r and z change, but the distance of passage p is constant. The rate of angular translation of an object in the image is $T = \dot{\phi}$. The pixel translation is approximately given by multiplying the angular translation by the focal length. By geometry of Figure 5.1 (a), we have:

$$z = p \cot \phi \tag{5.1}$$

To find the rate of translation $\dot{\phi}$, this expression is differentiated to get:

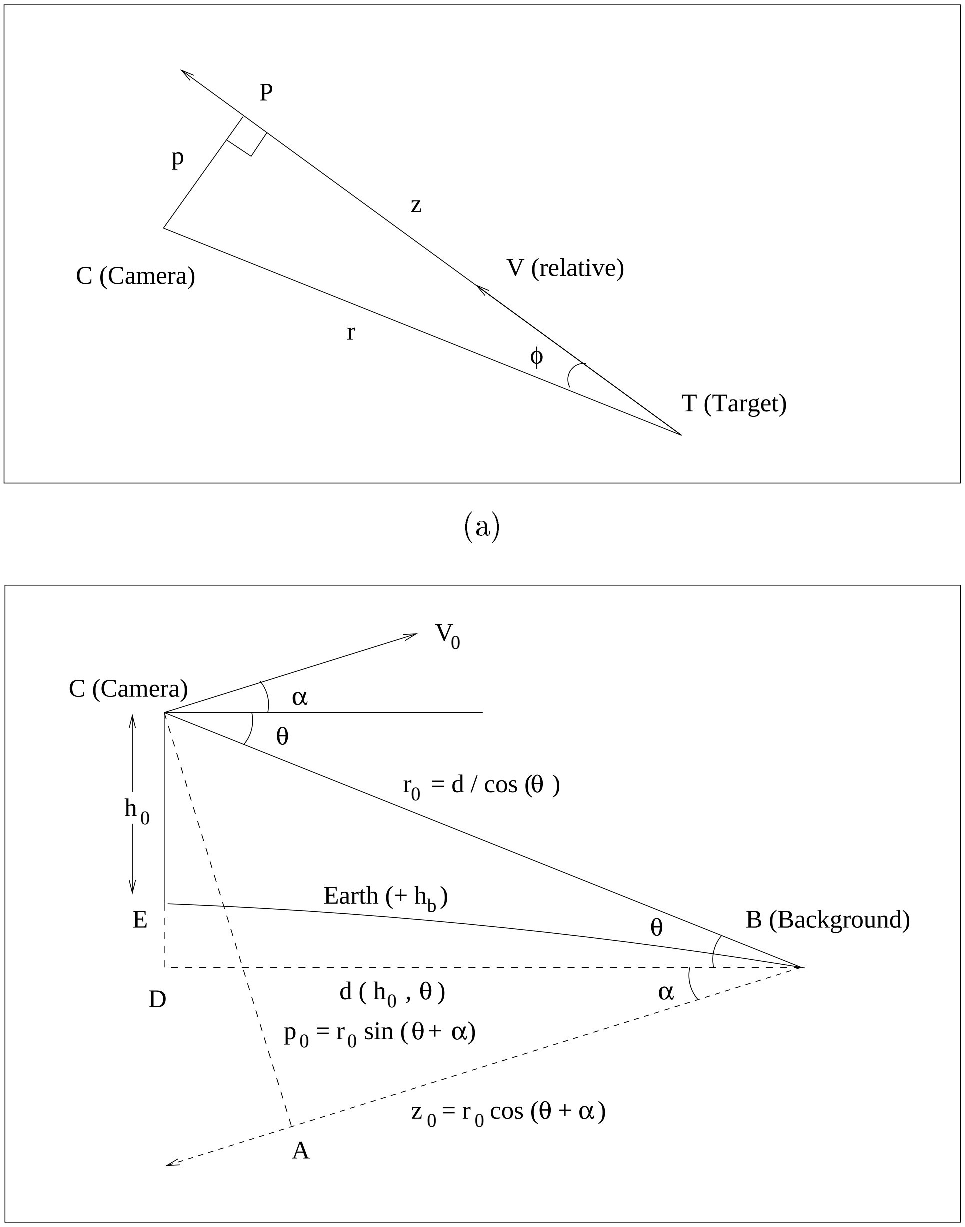
$$\dot{z} = -p(\csc^2 \phi)\dot{\phi} \tag{5.2}$$

The magnitude of the relative velocity V is the rate of decrease of z, given by:

$$V = -\dot{z} = p(r/p)^2\dot{\phi} \tag{5.3}$$

Also, the time of passage is given by:

$$\tau = z/V = r \cos \phi / V \tag{5.4}$$



(a)

(b)

Figure 5.1: Geometry of (a) target (b) background moving relative to the camera.

From equations (5.3) and (5.4), the rate of target translation is given by:

$$T = \dot{\phi} = \frac{pV}{r^2} = \frac{p\cos\phi}{\tau r} \tag{5.5}$$

Thus, the rate of image translation is proportional to the distance of passage, and the objects on a collision course are likely to have a smaller rate of translation compared to other objects. However, this rate is also dependent on the target distance, and a nearer target moves faster in the image than a farther target with the same distance of passage. If S_{min} is the smallest visible dimension that an object can have, the corresponding size in the image is given by:

$$s \geq s_{min} = S_{min}/r \tag{5.6}$$

Hence, from equation (5.5), one can write:

$$\frac{T}{s} \leq \frac{p\cos\phi}{\tau S_{min}} \leq \frac{p}{\tau S_{min}} \tag{5.7}$$

Hence, an object on a near collision course, having sufficient time before imminent collision has the ratio of its image motion to its image size bounded by the above pre-computable limit. For example, if the distance of passage of $p = 150\,m$ $(500\,ft)$ is allowed, and an object of smallest size of $S_{min} = 1.2\,m$ $(4\,ft)$ is to be detected before $\tau = 25$ seconds (750 frames), then this ratio becomes 1/6 – i.e., the image motion per frame is at the most $1/6^{th}$ of the image size of the object. However, in actual practice, a larger range of velocities should be checked, to have a safety margin.

It should be noted that the above relationship is valid only if the aircraft does not rotate or vibrate around its own axes. If there is rotation, it should be compensated by using the data from the aircraft navigation system. In the absence of this data, it may be possible to use image features due to clutter (if available) to perform the compensation, by modeling their image motion due to camera rotation.

If this compensation is successful, the velocity to size ratio of the object would be bounded. By reducing the image resolution to an appropriate level, the image velocity of the object would also be restricted. Hence, using pyramid construction, target detection can be performed at a number of resolutions, and the suitable resolution selected. This also leads to spatio-temporal integration of the image data and the amplification of SNR which could enable detection of sub-pixel or low-contrast objects in uniform background, such as clear or overcast sky.

The relationship between image motion and the distance of passage can be used to remove the clutter which is not on collision course and thus expected to have a large image motion.

However, the image motion is inversely proportional to the distance of the object from the camera. Thus, if clutter is at a large distance, it too could have a small image motion. The conditions under which an object on the collision course can be distinguished from ground clutter at the same image position are derived below.

Let r_0 and p_0 denote the background distance, and the minimum distance of approach for the background, respectively, as shown in Figure 5.1 (b). The relative velocity V_0 between the camera and the background is actually the magnitude of the camera velocity, By substituting these parameters in equation (5.5), the rate of background translation can be written as:

$$T_0 = \frac{p_0 V_0}{r_0^2} \tag{5.8}$$

Let $h_0 = h_c - h_b$ denote the difference between the camera altitude h_c and the background altitude h_b. Also, the angle of the camera velocity above the horizontal (not horizon) is α. From Figure 5.1 (b), we have:

$$\begin{aligned} r_0 &= d \sec\theta && (5.9) \\ p_0 &= r_0 \sin(\theta + \alpha) && (5.10) \end{aligned}$$

Here, d is a function of the relative height h_0 and the angle θ. If the earth were flat (or θ is large), refraction of light is negligible, and the terrain is smooth, the dotted line corresponding to d would coincide with the surface of the earth, and we would have

$$d = h_0 \cot\theta$$

However, if we express:

$$d(h_0, \theta) = h_0 \cot\theta f(h_0, \theta) \tag{5.11}$$

then the effects of the earth's curvature and refraction of light ray would be incorporated in the function f. If these factors can be neglected, then $f(h_0, \theta) \simeq 1$. The expression for f using the curvature of the earth is derived in Section 5.4. Also, using equation (5.9), one can write:

$$r_0 = h_0 \csc\theta f(h_0, \theta) \tag{5.12}$$

Substituting equations (5.10) and (5.12) in (5.8), the rate of background translation T_0 is given by:

$$T_0 = \frac{V_0 \sin(\theta + \alpha)}{r_0} = \frac{V_0 \sin(\theta + \alpha) \sin\theta}{h_0 f(h_0, \theta)} \tag{5.13}$$

If the hazard is to be discriminated from the background in the same line of sight, the rate of translation of the hazard must be much smaller than that of the background – i.e.,

$T \leq \eta_t^{-1} T_0$ with $\eta_t > 1$, having a larger value for greater discriminating power. Using equations (5.5) and (5.13), we have:

$$\frac{p \cos\phi}{\tau r} \leq \eta_t^{-1} \frac{V_0 \sin(\theta + \alpha) \sin\theta}{h_0 f(h_0, \theta)} \tag{5.14}$$

Hence, the object distance r should be larger than the following expression:

$$r \geq \frac{\eta_t p h_0 f(h_0, \theta) \cos\phi}{\tau V_0 \sin(\theta + \alpha) \sin\theta} = \frac{\eta_t p D f(h_0, \theta) \sqrt{1 - Q^2}}{\sin(\theta + \alpha) \sin\theta} \tag{5.15}$$

with

$$D = \frac{h_0}{\tau V_0} \,,\; Q = \frac{p}{r} = \sin\phi \,,\; \cos\phi = \sqrt{1 - Q^2} \simeq 1 \text{ (for } p \ll r) \tag{5.16}$$

Hence, θ should satisfy:

$$\sin(\theta + \alpha) \sin\theta \geq \eta_t D Q \sqrt{1 - Q^2} f(h_0, \theta) \tag{5.17}$$

Also, using $T \leq \eta_t^{-1} T_0$, with equations (5.5) and (5.13), one can write:

$$\frac{p \cos\phi}{\tau r} \leq \eta_t^{-1} \frac{V_0 \sin(\theta + \alpha)}{r_0} \tag{5.18}$$

Since the object distance cannot be greater than the background distance in the line of sight, $r \leq r_0$. Hence, one can also write:

$$\sin(\theta + \alpha) \geq \frac{\eta_t p \cos\phi}{\tau V_0} \frac{r_0}{r} \geq \frac{\eta_t p \sqrt{1 - Q^2}}{\tau V_0} \tag{5.19}$$

For $p \ll r$ or $Q \ll 1$, this condition is approximately independent of r. It can be said that for detection to be possible at all for a particular θ and α, the above condition is necessary irrespective of the target distance r, provided it is sufficiently large.

If the curvature of the earth and the refraction of light can be neglected, then $f \simeq 1$. The necessary condition in equation (5.19) does not simplify. However, equation (5.17) reduces to:

$$\sin(\theta + \alpha) \sin\theta \geq \eta_t D Q \sqrt{1 - Q^2} \tag{5.20}$$

On solving for θ, this yields:

$$\theta \geq \frac{1}{2} \left[\cos^{-1} \left(-2\eta_t D Q \sqrt{1 - Q^2} + \cos\alpha \right) - \alpha \right] \tag{5.21}$$

If $\alpha = 0$, the solution for θ is simpler:

$$\theta \geq \sin^{-1} \sqrt{\eta_t D Q \sqrt{1 - Q^2}} \tag{5.22}$$

For example, if we have:

$$p = 150\,m,\ \tau = 25\,s,\ V_0 = 150\,m/s,\ h_0 = 1\,km,\ \alpha = 0,\ \eta_t = 2.5 \tag{5.23}$$

For these values $D = 0.267$, and from equation (5.19) the necessary condition is $\theta \geq 5.7°$. This condition corresponds to the target being at the same position as the background, which is $r = r_0 = 10\,km \simeq 5.4\ nmi$ or $Q = 0.015$, using equation (5.12). However, if the target is nearer, the condition on θ is determined by equation (5.17) or (5.20). For example, if a hazard should be detected at $r = 5\,km \simeq 2.7\,nmi$ or $Q = 0.03$, one would really need $\theta \geq 8.1°$. The required θ increases as r decreases.

5.3 Detection using expansion

Another discriminating feature between objects on collision course, and objects much farther, is the time to collision. It is well known that the rate of image expansion, – i.e., the increase of the image size of an object – is inversely proportional to the time to collision.

In Figure 5.1 (a), as the object comes closer to the camera along the line of z, its size in the image will become larger. The rate of this expansion of any object is defined as the ratio of the rate of increase in its size to the size at that time, – i.e., $E = \dot{s}/s$ – where s is the size of the object in the image. Since $s = S/r$ where S is the object size which is assumed constant, we have $\dot{s} = -S\dot{r}/r^2$, and

$$E = -\dot{r}/r \tag{5.24}$$

By geometry of Figure 5.1 (a),

$$r^2 = z^2 + p^2 \tag{5.25}$$

To find the rate of expansion, this expression is differentiated to yield:

$$2r\dot{r} = 2z\dot{z} = -2zV \tag{5.26}$$

Hence, rate of target expansion is given by:

$$E = -\frac{\dot{r}}{r} = \frac{zV}{r^2} = \frac{V\cos\phi}{r} = \frac{\cos^2\phi}{\tau} \tag{5.27}$$

where the time to passage is:

$$\tau = z/V = r\cos\phi/V \tag{5.28}$$

For $\tau = 25\,s = 750$ frames, the ratio is 0.13 % per frame, which is a very small magnitude. This small expansion can be measured by tracking it over a large number of frames.

For estimating the rate of expansion of the background, the corresponding parameters for the background are substituted in equation (5.27) to give:

$$E_0 = \frac{z_0 V_0}{r_0^2} \tag{5.29}$$

Using $z_0 = r_0 \cos(\theta + \alpha)$ with equations (5.9) and (5.11), the rate of background expansion can be written as:

$$E_0 = \frac{V_0 \cos(\theta + \alpha)}{r_0} = \frac{V_0 \cos(\theta + \alpha) \cos\theta}{d} = \frac{V_0 \cos(\theta + \alpha) \sin\theta}{h_0 f(h_0, \theta)} \tag{5.30}$$

If reliable discrimination of the hazard from the background in the same line of sight is required, the rate of expansion of the hazard must be much larger than that of the background, – i.e., $E \geq \eta_e E_0$ with $\eta_e > 1$, having a large value for greater discriminating power. Using equations (5.27) and (5.30), one needs:

$$\frac{\cos^2\phi}{\tau} \geq \eta_e \frac{V_0 \cos(\theta + \alpha) \sin\theta}{h_0 f(h_0, \theta)} \tag{5.31}$$

or

$$\cos(\theta + \alpha) \sin\theta \leq \frac{h_0 f(h_0, \theta) \cos^2\phi}{\eta_e \tau V_0} = \eta_e^{-1} D(1 - Q^2) f(h_0, \theta) \tag{5.32}$$

where D and Q are given by equation (5.16). For the case of $f \simeq 1$, the equation (5.32) reduces to:

$$\cos(\theta + \alpha) \sin\theta \leq \eta_e^{-1} D(1 - Q^2) \tag{5.33}$$

Explicit solution for θ is then given by:

$$\theta \leq \frac{1}{2} \left[\sin^{-1} \left(2\eta_e^{-1} D(1 - Q^2) + \sin\alpha\right) - \alpha\right] \tag{5.34}$$

For the conditions in equation (5.23), we need $\theta \leq 6.2°$ for reliable detection using expansion.

It should be noted that the expansion in image size can also be caused by the rotation of the target aircraft in a way which would expose a larger area to the camera. However, this false expansion takes place only in the direction perpendicular to the axis of rotation of the target aircraft, whereas the expansion due to a potential collision would take place uniformly in all directions. Also, the target expansion will cease after the aircraft fully rotates to a position where maximum area is exposed to the camera. It may be possible to use these properties to discriminate between the false expansion and the expansion due to a collision course.

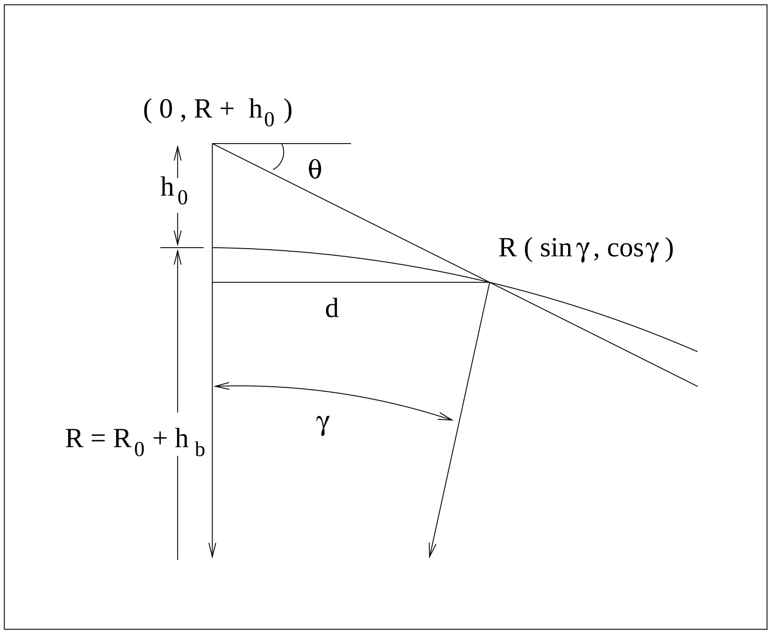


Figure 5.2: Geometry of earth's curvature: The coordinates used are with respect to earth's center.

5.4 Effect of horizon

In this section, function describing the effect of the curvature of the earth is calculated, neglecting the effects of refraction. Figure 5.2 shows the geometry of the earth's curvature. The coordinates used are with respect to earth's center. Using this, we have:

$$d = R\sin\gamma,\ d\tan\theta = h_0 + R(1-\cos\gamma) \simeq h_0 + d^2/(2R) \tag{5.35}$$

where $R = R_0 + h_b$, h_b is the altitude of the background, R_0 is the radius of earth, and γ is the angle subtended on the center of the earth by the triangle. Solving this equation yields:

$$d = R\left[\tan\theta \pm \sqrt{\tan^2\theta - 2h_0/R}\right] \tag{5.36}$$

The correct solution is the smaller value of d, since the larger value represents the other intersection of the line of sight with the earth.

$$d = R\left[\tan\theta - \sqrt{\tan^2\theta - 2h_0/R}\right] = \frac{2h_0}{\tan\theta + \sqrt{\tan^2\theta - 2h_0/R}} \tag{5.37}$$

By substituting in equation (5.11), we have:

$$f(h_0, \theta) = \frac{2}{1 + \sqrt{1 - 2h_0/(R\tan^2\theta)}} \tag{5.38}$$

If $\theta \simeq \pi/2$, or R is large, h is small, then $f \simeq 1$, – i.e., the earth's curvature can be neglected. However, where the line of sight just touches the earth – i.e., at the horizon – the discriminant under the square root is zero, then $f = 2$ and the corresponding θ is:

$$\theta_h = \tan^{-1}\sqrt{2h_0/R} \tag{5.39}$$

Any value of θ smaller than this value corresponds to the line of sight not touching the earth – i.e., background above the horizon.

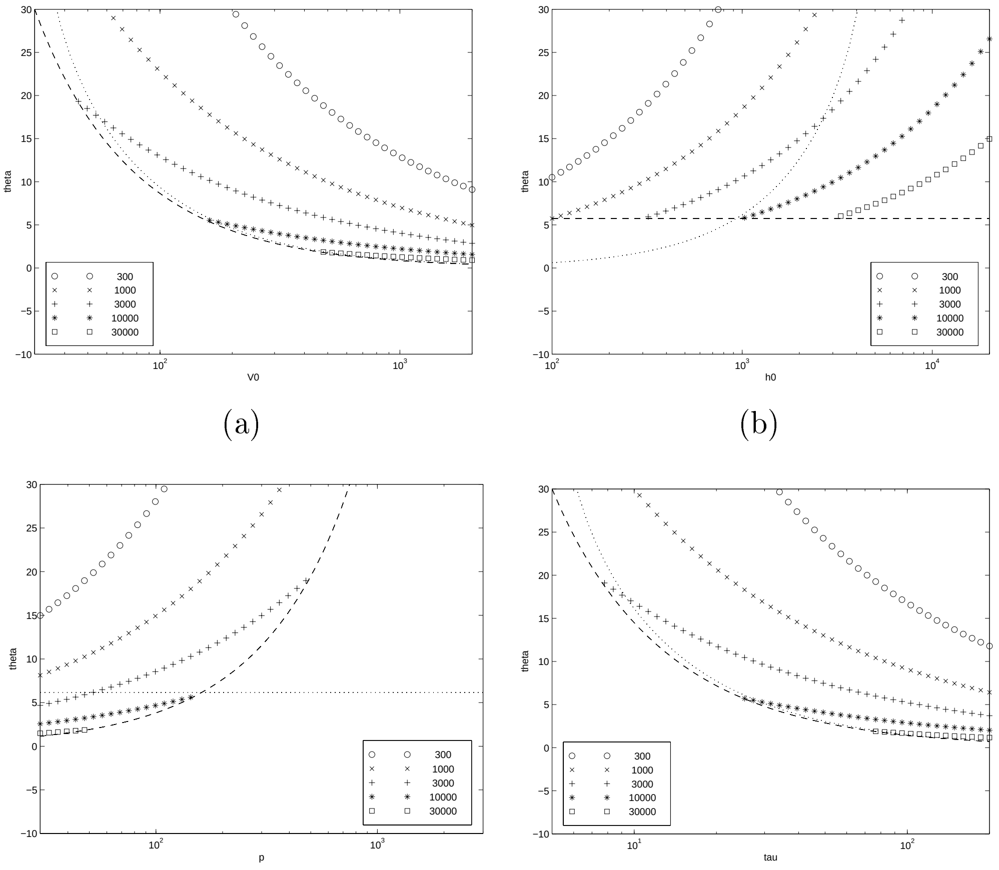
5.5 Behavior of translation and expansion

Figure 5.3 shows the variation of the required θ with the horizontal, for the possibility of detection using translation and expansion, against various parameters. Effect of horizon was neglected since it was observed that it does not affect the plots to a significant extent. The minimum θ for detection using translation, which is shown by dashed line, whereas the maximum θ for detection using expansion is shown by dotted line. However, the minimum θ criterion is only the necessary criterion. For actual discrimination using translation for an object at a given distance, a larger θ is required. The other curves show the required θ for detection using translation for various object distances in meters, and are enveloped by the dashed line curve.

Most of the information in these curves can be condensed using the parameter $D = h_0/(\tau V_0)$. Figure 5.4 (a) shows the contours of same D for different values of V_0 and h_0 for $\tau = 25\,s$. Plots of required θ for translation and expansion using a number of values of the target distance r in km, for the distance of passage $p = 150\,m$ are shown in Figure 5.4 (b). However, the necessary criterion for translation cannot be expressed using these plots.

5.6 Estimation of translation and expansion

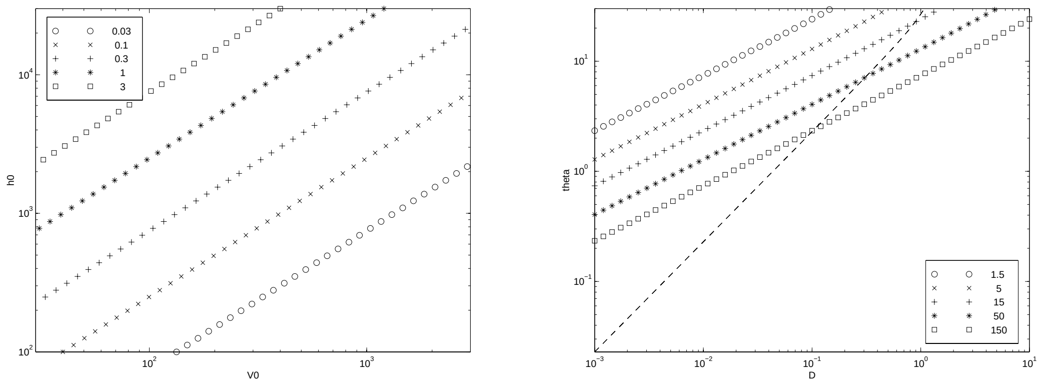
To reduce the computational complexity of estimating the translation and expansion, a feature-based approach was used. A morphological filter [6] which subtracts the opening and closing of the image from the original image was used to detect positive and negative features, corresponding to light and dark objects, respectively.



(a) (b)

(c) (d)

Figure 5.3: Variation of the required θ with the horizontal, for the possibility of detection using translation and expansion, against a number of parameters: (a) Camera velocity: V_0, (b) Relative height between camera and background: h_0, (c) Distance of passage: p (d) Time of passage (or collision): τ. Default values of the parameters (except when they vary) are: $V_0 = 1\,km/s$, $h_0 = 1\,km$, $p = 150\,m$, and $\tau = 25\,s$, and $\eta_t = \eta_e = 2.5$. The minimum θ for detection using translation is shown by dashed line, whereas the maximum θ for detection using expansion is shown by dotted line. The other curves show the required θ for translation for various object distances in meters.



(a) (b)

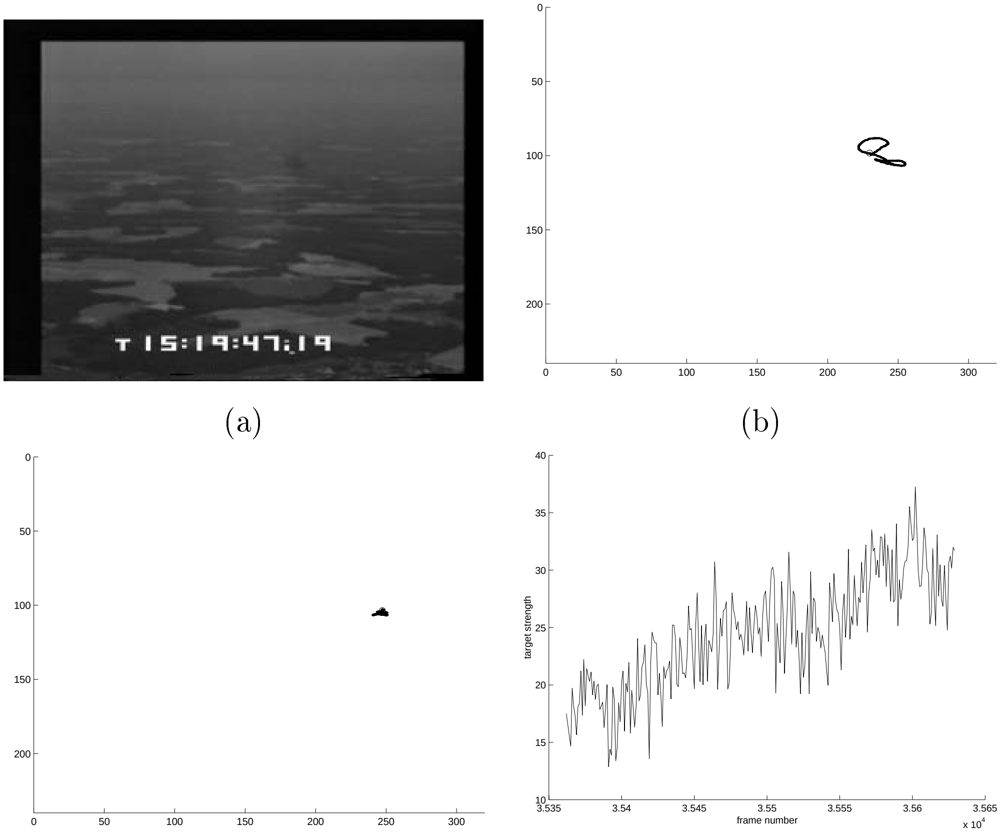
Figure 5.4: Plots for detection using translation and expansion: (a) Plot showing the contours of same D for different values of V_0 and h_0 for $\tau = 25\,s$ (b) Plots of θ required for detection using translation are shown with various symbols for a number of values of the target distance r in km, for the distance of passage $p = 150\,m$. Plot of the θ required for detection using expansion is shown with a dashed line.

To estimate the translation of the features over a number of frames, they were tracked over a number of frames. In case of navigation system data being available, the position of the features were compensated before performing the tracking. A nearest neighbor approach was used to determine the corresponding feature in the next frame, and the smoothed estimates of the feature position and velocity in each frame were obtained using Kalman filter approach. This procedure is similar to the one described in Chapter 7 used for detecting targets crossing the aircraft.

For detecting expansion, a 15 × 15 window around each feature was explored. The sub-image corresponding to the window was thresholded, and the connected component containing the center of the window was found. All the pixels in the sub-image that did not belong to the component were set to zero. To estimate the size of the component, the sub-image was convolved with a number of smoothing masks. These masks perform matched filtering with a object templates corresponding a number of different sizes. The maximum output from all these masks was considered as the measure of target strength. The rate of expansion was measured in terms of increase of the target strength, tracked over a number of frames. The target strength was plotted against the frame number, and the rate of expansion was estimated by applying least squares to the logarithm of the target strength.

5.7 Results

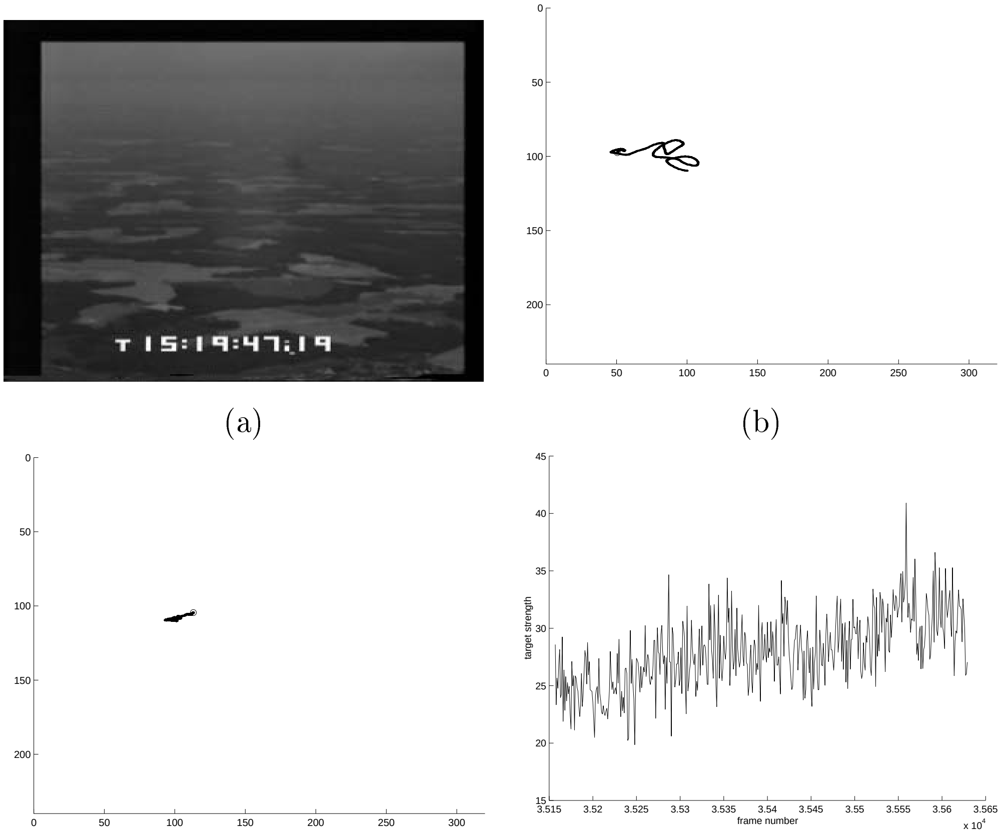
The estimation of translation and expansion was performed on a sequence of images captured from an analog camera in which the target aircraft is approaching the camera. Figure 5.5 (a) shows a typical frame from the sequence. Figure 5.5 (b) and (c) show the target track in the original and the motion compensated images, respectively. Figure 5.5 (d) shows the plot of the estimated target size against the frame number. Corresponding plots for two clutter tracks are shown in Figures 5.6 and 5.7. It can be seen that the target expansion is the large for the target track, and small for the clutter tracks. On the other hand, the rate of target translation is small for the target track and large for the clutter tracks. Figure 5.8 shows the significant tracks before and after motion compensation. A scatter plot of the feature expansion against translation for these tracks, including the target track is shown in Figure 5.9. The rate of translation is measured in terms of the displacement magnitude of the compensated features in 100 frames, whereas the rate of expansion is measured in terms of the increase in the logarithm (to base 10) of the target strength in 100 frames. It is seen that the target has a large rate of expansion and a small rate of translation and is located in the upper left corner of the scatter plot.



(a) (b)

(c) (d)

Figure 5.5: Translation and expansion for target track: (a) Sample image from the last frame. (b) Target track (c) Target track after compensation. Rate of translation is small for target track. (d) Plot of expansion against frame number. Rate of expansion is large for target track.



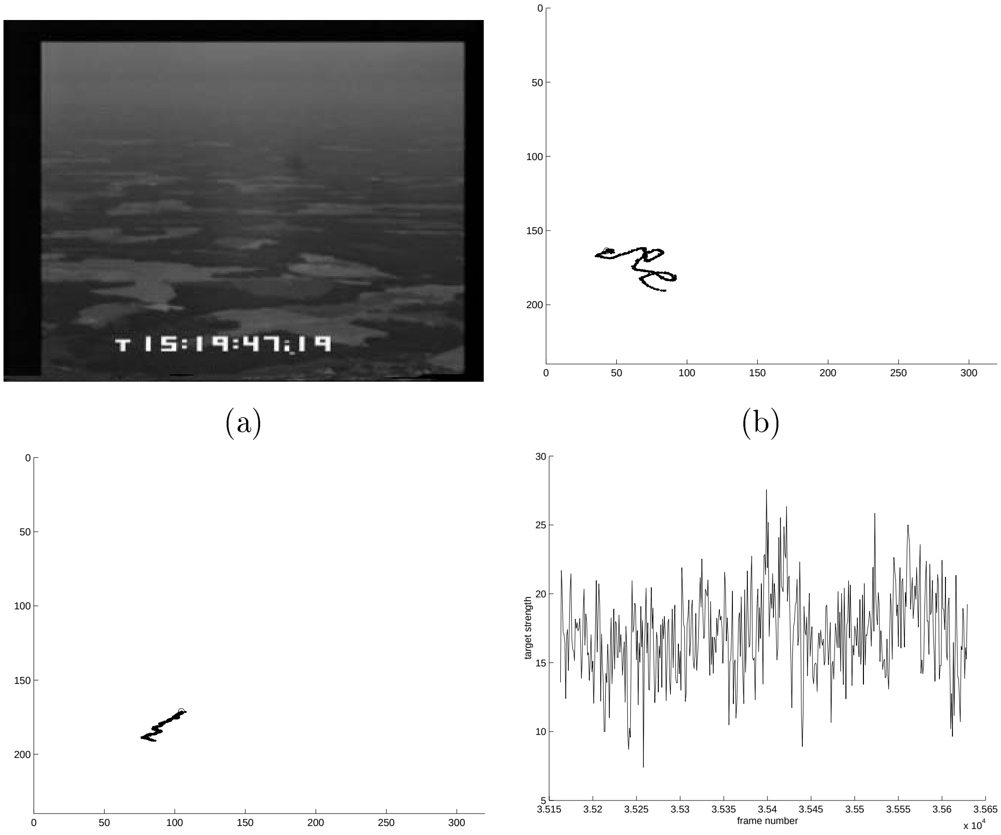
(a)

(b)

(c)

(d)

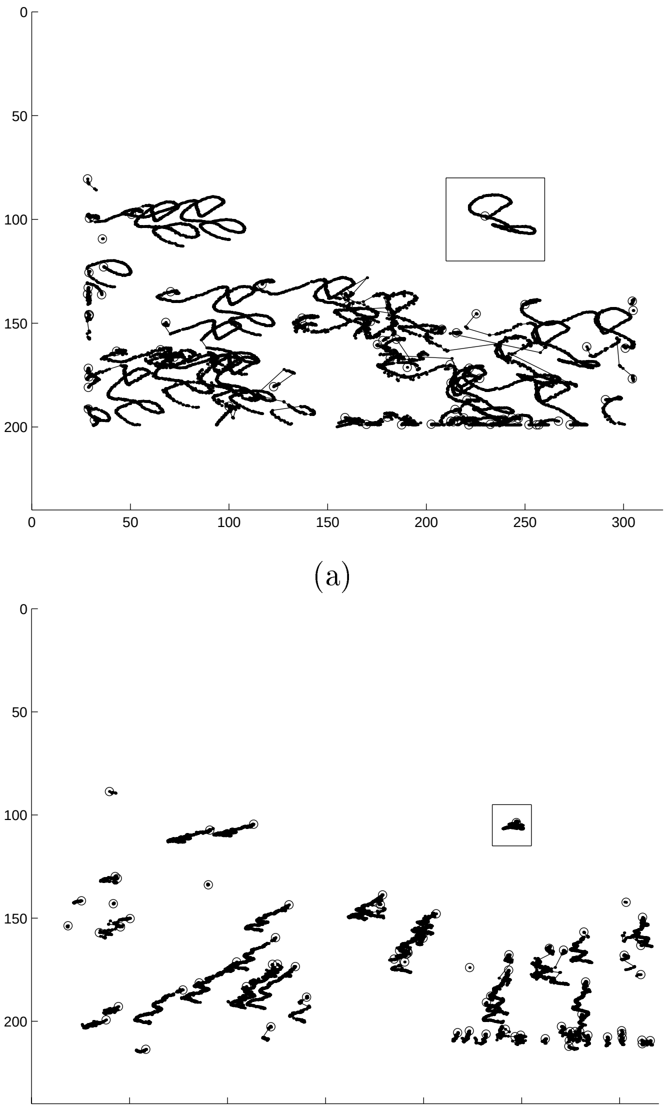
Figure 5.6: Translation and expansion for clutter track: (a) Sample image from the last frame. (b) Target track (c) Target track after compensation. Rate of translation is large for clutter track. (d) Plot of expansion against frame number. Rate of expansion is small for clutter track.



(a) (b)

(c) (d)

Figure 5.7: Translation and expansion for another clutter track: (a) Sample image from the last frame. (b) Target track (c) Target track after compensation. Rate of translation is large for clutter track. (d) Plot of expansion against frame number. Rate of expansion is small for this clutter track.



(a)

(b)

Figure 5.8: Feature tracks (a) before, and (b) after rotation compensation: Target track surrounded by a rectangle has a small translation after compensation.

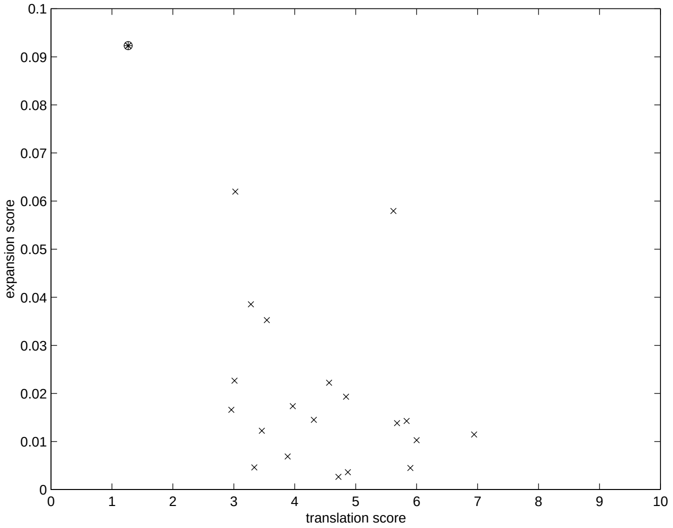


Figure 5.9: Scatter plot of the feature expansion against translation: The rate of translation is measured in terms of the displacement magnitude of the compensated features in 100 frames, whereas the rate of expansion is measured in terms of the increase in the logarithm (to base 10) of the target strength in 100 frames. The target is marked as an encircled asterisk, and is in upper left corner, having a small rate of translation and a large rate of expansion.

Chapter 6

Algorithm Fusion

Each of the target detection algorithms has its own advantages and limitations. Hence, a combination of these algorithms may be used in the ultimate design to overcome their individual limitations while maximizing their advantages. This chapter describes a method for combining the algorithms using statistical approach to optimize the performance in terms of the mis-detection and false alarm rates. In particular, the pre-processing algorithms of low-stop and morphological filters, described in Chapter 2 are combined. The performance of the fused algorithm is compared with the original algorithms using the methodology described in Chapter 3.

6.1 Combination of algorithms using a statistical approach

According to the Neyman Pearson criterion, the optimal Bayesian detector which minimizes the rate of mis-detection for a particular rate of false alarms is obtained by thresholding the joint likelihood ratio of the individual detector outputs, or some monotonic function of the same. The threshold should be such that the desired false alarm rate is obtained.

Consider the joint likelihood ratio of the low-stop and the morphological filter. Let $z = (z_1, z_2)$ be the 2-D vector denoting the outputs of the low-stop and the morphological filters, respectively. Let $p(z|H_0, C)$ and $p(z|H_1, C)$ denote the joint probability density functions for the hypotheses denoting the absence and presence of a target, respectively, for clutter level estimate C. The likelihood ratio is then given by:

$$L_{H,C}(z) = \frac{p(z|H_1, C)}{p(z|H_0, C)} \tag{6.1}$$

6.2 Statistical behavior of low-stop and morphological filters

In the following analysis, it is assumed that the input image pixels are described by the sum of the signal θ, background level β, and the camera noise ν, which is modeled as an uncorrelated Gaussian noise of zero mean and variance η^2.

$$x = \theta + \beta + \nu \tag{6.2}$$

If there is no clutter, the distributions of x in absence and presence of the target are given by:

$$p(x|H_0) \sim N(\beta, \eta^2) \text{ , } p(x|H_1) \sim N(\theta + \beta, \eta^2) \tag{6.3}$$

If clutter is present, the exact distributions would depend on the nature of the clutter. Here, it is assumed that the presence of clutter changes the mean background level, and the variance parameter of the noise, making these parameters space varying.

Low-stop filtering is performed by subtracting the low-pass filtered image, using a weighted spatial average of the neighborhood, from the original image. This filter attempts to subtract the background level. Since it is a linear filter, if the input is normally distributed, the output z_l will also be distributed as:

$$p(z_l|H_0) \sim N(0, \sigma_l^2) \text{ , } p(z_l|H_1) \sim N(\mu_l, \sigma_l^2) \tag{6.4}$$

with

$$\sigma_l = f_l \eta \text{ , } \mu_l = g_l \theta \tag{6.5}$$

where f_l and g_l are the amplification gains in the standard deviation and mean due to the filter. It should be noted that the background level β is subtracted out by the filter.

Morphological filtering is performed by taking the difference between the original image and its opening (positive targets) or closing (negative targets). Without loss of generality, only positive targets are considered, which are detected by subtracting the opening from the original image. This is expected to remove uniform background, as well as most of the clutter.

To obtain a model for the distribution of the morphological filter and to verify the distribution of low stop filter, simulations were performed. A large number of floating point images containing Gaussian noise were generated. Low-stop and morphological filter were applied to these images, and the histograms of the filter outputs were obtained. Figure 6.1 (a) shows the histogram of the original image with Gaussian noise. Figure 6.1 (b) shows the histogram of the low-stop filter output, which is normally distributed with zero mean, as expected. The histogram of the morphological filter output is shown in Figure 6.1 (c). It can be seen that the histogram resembles a normal distribution with a positive mean. However, since the opening of an image is always less than or equal to the original image, the filter output is always non-negative. Hence, the distribution is truncated on the negative side, and has an impulse at zero in place of the negative values. For clarity, the distribution after removing the impulse is shown in Figure 6.1 (d).

This distribution can be modeled by using a hypothetical normally distributed variable $\xi_m \sim N(\mu_m, \sigma_m^2)$. The output z_m of the morphological filter can be expressed in terms of ξ_m as:

$$z_m = \max(\xi_m, 0) \tag{6.6}$$

It can be shown that the explicit distribution of z_m is given by:

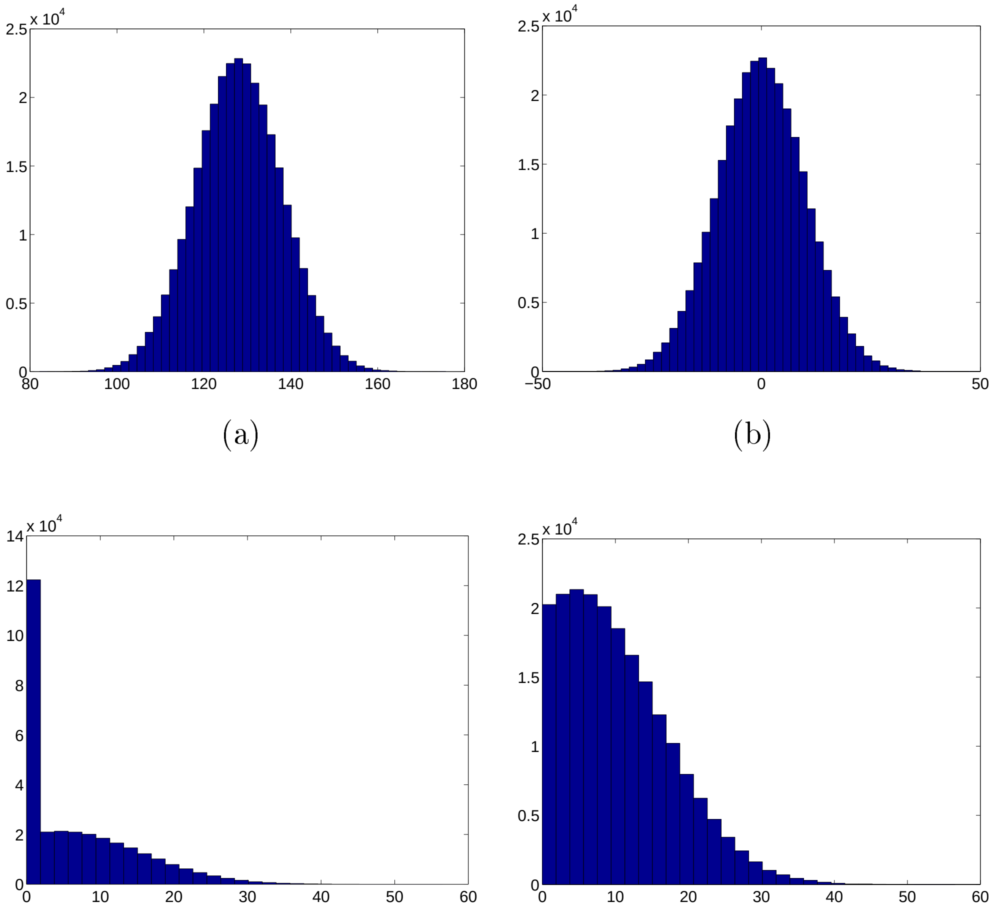
$$p(z_m|H_0) = \frac{u(z_m)}{\sigma_m} G\left(\frac{z_m - \mu_m}{\sigma_m}\right) + \delta(z_m)\, \Phi\left(-\frac{\mu_m}{\sigma_m}\right) \tag{6.7}$$

where $u(\cdot)$ is the unit step function, $\delta(\cdot)$ is the Dirac impulse function, and $G(\cdot)$ and $\Phi(\cdot)$ are the probability density and cumulative distribution functions of a standard normal variable, respectively. It can be shown that the mean and variance of this distribution, which are different from the parameters μ_m and σ_m^2, can be expressed as:

$$\begin{aligned} m_m &= \mu_m \Phi(\mu_m/\sigma_m) + \sigma_m G(\mu_m/\sigma_m) \\ s_m^2 &= \sigma_m^2 \Phi(\mu_m/\sigma_m) - m_m(m_m - \mu_m) \end{aligned} \tag{6.8}$$

Hence, the parameters μ_m and σ_m can be obtained from the observed values of m_m and s_m^2 by using a numerical method. It can be shown that this procedure yields the maximum likelihood estimates of the parameters. The parameters derived from the above simulations are shown in Table 6.1.

To obtain the distribution in presence of a target, a number of simulated targets of fixed amplitude were added to each of the images generated above. Morphological filter was applied to these images, and a histogram of pixel values only at the target positions was obtained. However, since the number of targets is not as large as the total number of pixels in the image, the histogram is less reliable than in the case of absence of targets. These experiments were repeated for various signal amplitudes and the sample mean and variance of the outputs were computed. The sample means and variances were taken as the estimates



(a)

(b)

(c)

(d)

Figure 6.1: Statistics of low-stop and morphological filters: Histograms of: (a) Input image with Gaussian noise. (b) Output of low-stop filter. (c) Output of morphological filter. (d) Output of morphological filter after removing impulse at zero value.

Table 6.1: Statistical parameters of low-stop and morphological filters derived from simulations

Parameter	Value
η	10.0
$m_l = \mu_l$	-6.6e-08 $\simeq$ 0.0
$s_l = \sigma_l$	9.9815 $\simeq$ 10.0
m_m	7.0539
s_m	7.8352
μ_m	4.6293
σ_m	10.8423

of the means and variances of the distributions. For the low-stop filter, the parameters μ_l and σ_l coincide with the distribution mean and variance m_l and s_l^2, respectively, and are approximately equal to the signal amplitude θ and the input noise standard deviation η, respectively, corresponding to $g_l \simeq 1$ and $f_l \simeq 1$. For the morphological filter, the actual parameters μ_m and σ_m of the underlying normal distribution were calculated from the m_l and s_l^2 using the simultaneous equations (6.8). It was observed that the parameter σ_m is approximately equal to the noise intensity η, and does not change much with the signal amplitude θ. However, the parameter μ_m increases non-linearly with θ. It has a positive value at $\theta = 0$ – i.e., noise-only condition – and increases with a lower rate than the corresponding low-stop filter parameter μ_l. Figure 6.2 shows the plots of the parameters μ_l and μ_m against the signal amplitude θ.

The output of the morphological filter is invariant to the constant background level β. Furthermore, it also suppresses the clutter. Hence, the effective 'noise' intensity for the morphological filter would be different from that for the low-stop filter in case of cluttered scenario, and is denoted by η_m. However, in the case of the above simulations it is the same as the original noise intensity η. If η_m as well as the signal amplitude θ are scaled by a constant factor, σ_m and μ_m will get scaled by the same factor. Hence, outputs of the morphological filter for any general η_m can be written as:

$$\sigma_m = \eta_m \, f_m \ , \ \mu_m = \eta_m \, g_m \left(\frac{\theta}{\eta_m} \right) = \frac{\sigma_m}{f_m} \, g_m \left(\frac{f_m \theta}{\sigma_m} \right) \tag{6.9}$$

where f_m is the gain in standard deviation ($f_m \simeq 1$), neglecting the dependency on the target

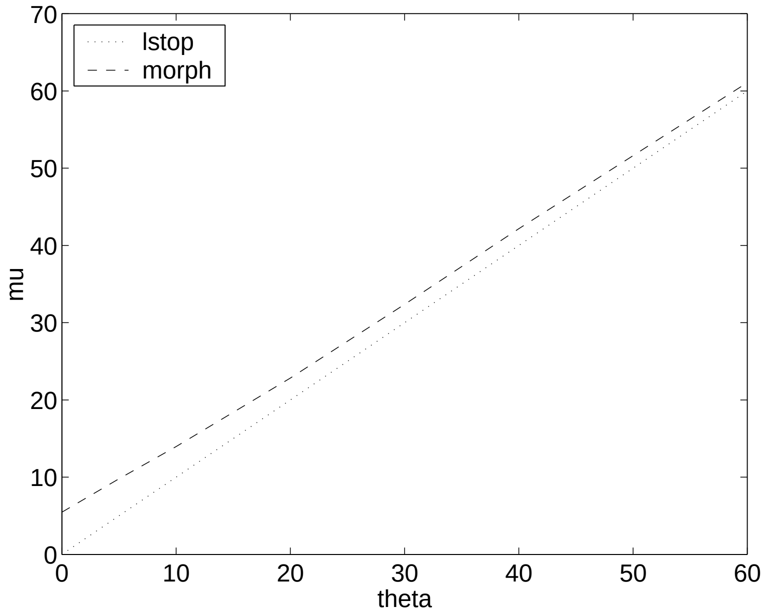


Figure 6.2: Plot of parameters μ_l and μ_m against signal amplitude θ for $\eta = 10$.

SNR, and $g_m(\cdot)$ is the gain in mean, depending on the target SNR θ/η_m. The function f_m can be obtained from the experimentally determined values of μ_m and σ_m for $\eta = 10$, plotted in Figure 6.2.

It was also observed that there is a correlation between the outputs of the low-stop and the morphological filters. Hence, the joint distribution of the two outputs is modeled as a normal distribution, truncated for the morphological filter. Assuming a hypothetical random variable $\xi = (\xi_l, \xi_m)^t$ which is normally distributed, the actual output vector z can be expressed as:

$$z = (z_l, z_m)^t = (\xi_l, \max(\xi_m, 0))^t \tag{6.10}$$

The parameters of distribution of z are:

$$\mu = \begin{bmatrix} \mu_l \\ \mu_m \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_l^2 & \rho\sigma_l\sigma_m \\ \rho\sigma_l\sigma_m & \sigma_m^2 \end{bmatrix} \tag{6.11}$$

where ρ is the correlation coefficient, and σ_l^2 and σ_m^2 are the individual variances of z_l and z_m, respectively. The distribution mean and covariance matrix are given by:

$$m = \begin{bmatrix} m_l \\ m_m \end{bmatrix}, S = \begin{bmatrix} s_l^2 & \rho' s_l s_m \\ \rho' s_l s_m & s_m^2 \end{bmatrix} \tag{6.12}$$

Note that due to the linearity of the low-stop filter, we have $m_l = \mu_l, s_l = \sigma_l$.

However, using these relations, it is analytically difficult to calculate the actual correlation coefficient parameter ρ from the observed correlation coefficient ρ'. Furthermore, such a computation would have to be repeated for every pixel, which is highly inefficient. Hence, the value of $\rho = \rho'$ is currently being used.

Using the above models of low-stop and morphological filter outputs, the distribution of z for $z_m > 0$ is given by:

$$\begin{aligned} p(z|H_0) &= |2\pi\Sigma|^{-1/2} \exp\left[(z-\mu_0)^t\Sigma^{-1}(z-\mu_0)/2\right] \\ p(z|H_1) &= |2\pi\Sigma|^{-1/2} \exp\left[(z-\mu_\theta)^t\Sigma^{-1}(z-\mu_\theta)/2\right] \end{aligned} \tag{6.13}$$

For $z_m < 0$, $p(z|H_i) = 0$ Also, there is an impulse function at $z_m = 0$, so that the integral of p becomes unity.

6.3 Bayesian fusion of multiple filters

The combined likelihood ratio of the two filters is given by:

$$L_{H,C}(z) = \frac{p(z|H_1, C)}{p(z|H_0, C)} \simeq \frac{N(\mu_\theta, \Sigma_C)}{N(\mu_0, \Sigma_C)} \tag{6.14}$$

where μ_θ and μ_0 are 2-D vectors denoting the mean outputs of the algorithms in presence and absence of target. The covariance matrix Σ_C, which depends on the clutter level, can be estimated using the image, and will be denoted by Σ for brevity. The same covariance is used for the presence and absence of the target, since it is experimentally observed that there is not much difference between the respective covariances.

Using equation (6.13), the log likelihood ratio (LLR) is given by:

$$\begin{aligned} l(z) &= \log L_{H,C}(z) = -\frac{1}{2}(z-\mu_\theta)^t\Sigma^{-1}(z-\mu_\theta) + \frac{1}{2}(z-\mu_0)^t\Sigma^{-1}(z-\mu_0) \\ &= (\mu_\theta-\mu_0)^t\Sigma^{-1}(z-\mu_0) - \frac{1}{2}(\mu_\theta-\mu_0)^t\Sigma^{-1}(\mu_\theta-\mu_0) \end{aligned} \tag{6.15}$$

The parameters of the LLR in absence of target – i.e., $E[z|H_0] = \mu_0$, $V[z|H_0] = \Sigma$ – can be computed as:

$$\begin{aligned} E[l(z)|H_0] &= (\mu_\theta-\mu_0)^t\Sigma^{-1}E[z-\mu_0|H_0] - \frac{1}{2}(\mu_\theta-\mu_0)^t\Sigma^{-1}(\mu_\theta-\mu_0) = -\frac{1}{2}d^2 \\ V[l(z)|H_0] &= (\mu_\theta-\mu_0)^t\Sigma^{-1}V[z-\mu_0|H_0]\Sigma^{-1}(\mu_\theta-\mu_0) = d^2 \end{aligned} \tag{6.16}$$

where d known as the deflection coefficient [13] is the generalization of the signal to noise ratio for multiple dimensions.

$$d = \sqrt{(\mu_\theta - \mu_0)^t \Sigma^{-1} (\mu_\theta - \mu_0)} \tag{6.17}$$

When the target of any strength is present, the variance parameter still remains the same but the mean parameter changes. For the target strength such that $E[z|H_1] = \mu_\theta$, the LLR parameters are given by:

$$\begin{aligned} E[l(z)|H_1] &= (\mu_\theta - \mu_0)^t \Sigma^{-1} E[z - \mu_0|H_1] - \frac{1}{2}(\mu_\theta - \mu_0)^t \Sigma^{-1} (\mu_\theta - \mu_0) = \frac{1}{2}d^2 \\ V[l(z)|H_1] &= (\mu_\theta - \mu_0)^t \Sigma^{-1} V[z - \mu_\theta|H_1] \Sigma^{-1} (\mu_\theta - \mu_0) = d^2 \end{aligned} \tag{6.18}$$

It is seen that the mean and variance of the LLR are dependent on the mean and variance parameters of the filter outputs. Due to this, the probability of false alarm and mis-detection also depends on these parameters. Accordingly, two approaches of obtaining a detector are shown below.

6.3.1 Constant False Alarm Rate (CFAR) detector

To get a constant false alarm rate irrespective of the local variance, the LLR is normalized so that it would have a zero mean and unit variance in absence of the target. The resulting function is given by:

$$D(z) = \frac{l(z) - E[l(z)|H_0]}{\sqrt{V[l(z)|H_0]}} = \frac{(\mu_\theta - \mu_0)^t \Sigma^{-1} (z - \mu_0)}{\sqrt{(\mu_\theta - \mu_0)^t \Sigma^{-1} (\mu_\theta - \mu_0)}} \tag{6.19}$$

This is a matched filter, which matches the 2-D outputs from low-stop and morphological filters, to the expected outputs of these filters. Since $D(z|H_0) \sim N(0,1)$, if a threshold τ is applied, the false alarm rate is given by:

$$P_{FA} = 1 - \Phi\left(\frac{\tau - E[D(z)|H_0]}{V[D(z)|H_0]}\right) = 1 - \Phi(\tau) \tag{6.20}$$

where $\Phi(\cdot)$ denotes the cumulative distribution of a standard normal variable. Note that this is now independent of any parameters. In presence of a target so that $E[z|H_1] = \mu_\theta$, it can be easily seen that $D(z) \sim N(d, 1)$. Hence, the mis-detection rate is given by:

$$P_{MD} = \Phi\left(\frac{\tau - E[D(z)|H_1]}{V[D(z)|H_1]}\right) = \Phi(\tau - d) \tag{6.21}$$

The CFAR approach attempts to maintain a constant false alarm rate all over the image, irrespective of the local variance. Hence, it would be useful if a constant false alarm rate is required in all parts of the image, for example, if the parts are processed separately on parallel processors. To check the conditions under which this filter is optimal, the log likelihood ratio $l(z)$ is written in terms of the discriminant function $D(z)$ as:

$$l(z) = d\,D(z) - d^2/2 \tag{6.22}$$

It can be seen that, $l(z)$ and $D(z)$ are monotonic to each other when the deflection coefficient d, given by equation (6.17) remains constant. Under such conditions, thresholding $D(z)$ is equivalent to thresholding $l(z)$, the latter being the Bayesian optimum. The deflection coefficient is dependent on the covariance of the noise, as well as the target strength, and is the generalization of SNR for multiple dimensions. Thus, if the variance parameters of the individual filter outputs, as well the target amplitudes, are constant across the image, this approach is optimal in terms of the false alarms and mis-detection rates. However, in practice, the parameters (especially the low-stop filter output variance) do depend on the clutter level. In such a case, if the target amplitude is constant throughout the image, the CFAR approach is not optimal. However, if the criterion for good detection is to detect targets having a particular SNR – i.e., stronger targets in cluttered regions but weaker targets in uncluttered regions – the CFAR approach can be considered optimal.

It can be seen that $D(z)$ is dependent on the target amplitude θ through $\mu_\theta - \mu_0$, as well as d. If $\mu_\theta - \mu_0$ is a linear function of the target amplitude θ, it would cancel out in equation (6.19) and $D(z)$ would become independent of the signal amplitude θ. However, if $\mu_\theta - \mu_0$ is non-linear, the filter would be optimal only under specific conditions.

The false alarm rate is determined by the threshold τ, whereas the mis-detection rate is also determined by the deflection coefficient d. Consider optimizing the matched filter for a particular d, in an environment with clutter covariance Σ. If θ is the signal amplitude, equations (6.5) and (6.9) yield:

$$\begin{aligned}
\mu_\theta - \mu_0 &= \begin{bmatrix} \mu_{l\theta} - \mu_{l0} \\ \mu_{m\theta} - \mu_{m0} \end{bmatrix} = \begin{bmatrix} g_l\theta - 0 \\ \left(g_m(\frac{\theta}{\eta_m}) - g_m(0)\right)\eta_m \end{bmatrix} \\
&= \begin{bmatrix} g_l\theta \\ \frac{\sigma_m}{f_m}\left(g_m(\frac{f_m\theta}{\sigma_m}) - g_m(0)\right) \end{bmatrix}
\end{aligned} \tag{6.23}$$

Using this expression, the following equation should be numerically solved for θ by evaluating $\mu_\theta - \mu_0$ using equation (6.23) with the particular d.

$$(\mu_\theta - \mu_0)^t \Sigma^{-1} (\mu_\theta - \mu_0) = d^2 \tag{6.24}$$

However, if the covariance matrix Σ varies throughout the image, this procedure would have to be carried out for all pixels, which would be highly inefficient. Furthermore, the procedure optimizes only for a particular value of d.

Alternatively, if one assumes that d and θ are small, one can optimize the fusion using a Locally Most-Powerful (LMP) test [13, 17]. For small value of θ, we have:

$$\mu_\theta - \mu_0 \simeq \left(\frac{\partial \mu}{\partial \theta}\right)_{\theta=0} \cdot \theta = \begin{bmatrix} g_l \\ g'_m(0) \end{bmatrix} \theta = s\,\theta \tag{6.25}$$

where s is 2-D vector independent of θ. The expression is now linear in θ, and the discriminant function $D(z)$ becomes independent of θ.

$$D(z) = \frac{s^t \Sigma^{-1}(z - \mu_0)}{\sqrt{s^t \Sigma^{-1} s}} \tag{6.26}$$

with

$$s = \begin{bmatrix} g_l & g'_m(0) \end{bmatrix}^t \tag{6.27}$$

6.3.2 Direct thresholding of Log Likelihood Ratio (LLR)

As shown in the previous section, if the amplitude of the signal to be detected is fixed irrespective of the local variance, the overall mis-detection rate for a given overall false alarm rate is not minimized by the CFAR approach. In fact, there cannot be a single optimal detector for all amplitudes. Hence, the fusion should be optimized for a particular amplitude. A criterion for choosing this amplitude is described below.

Suppose that some particular minimum rates of false alarms as well as mis-detections are required for the algorithm. The amplitude corresponding to the minimum possible variance – i.e., the variance of the camera noise without clutter – can be used to tune the fusion. If the actual amplitude is smaller than this amplitude, even an optimal detector tailored to that amplitude will not give the required false alarm and mis-detection rates. On the other hand, since the performance of the detector increases monotonically with the amplitude, a larger amplitude yields a better performance, though it may not be optimal.

Suppose the LLR threshold is τ. Using the mean and the variance of the LLR in absence and presence of the target, given by equations (6.16) and (6.18), the false alarm and mis-detection rates can be computed as:

$$P_{FA} = 1 - \Phi\left(\frac{\tau + d^2/2}{d}\right)\ ,\ P_{MD} = \Phi\left(\frac{\tau - d^2/2}{d}\right) = 1 - \Phi\left(\frac{d^2/2 - \tau}{d}\right) \tag{6.28}$$

If one denotes:

$$\phi_0 = \Phi^{-1}(1 - P_{FA}) = \frac{\tau + d^2/2}{d} \, , \; \phi_1 = \Phi^{-1}(1 - P_{MD}) = \frac{d^2/2 - \tau}{d} \tag{6.29}$$

then τ can be eliminated to obtain:

$$\phi_0 - \phi_1 = d = \sqrt{(\mu_\theta - \mu_0)^t \Sigma^{-1} (\mu_\theta - \mu_0)} \tag{6.30}$$

The target amplitude can be chosen such that μ_θ corresponding to it satisfies this equation, using Σ^{-1} under noise only conditions.

6.4 Application on images

To apply this procedure on images, the statistical parameters are computed in an annular 31×31 window around each pixel, where an 11×11 window immediately around the pixel is excluded to reduce the biasing of parameters when the target is present at the pixel. There is a trade-off between using larger sized window giving more reliable estimates, and smaller sized window giving better localization in case of space varying clutter intensity. The window size used here was arbitrary. However, use of different window sizes can be explored to find the optimum window size.

Efficient methods are used to estimate the distribution mean m and the covariance S at each pixel of the low-stop and morphological output images. From these, the estimates of μ and Σ are calculated and stored as images. However, in some experiments, fixed values of μ_m and σ_m^2 were used for the morphological filter, since the estimates are less reliable, but do not change much over the image (unlike low-stop filter, where these parameters heavily depend on the clutter). The template signal for the matched filter is calculated using equation (6.27), and the matched filter is applied separately to each pixel.

6.5 Results

The algorithm fusion approach was evaluated using the performance characterization approach of Chapter 3. Background images obtained from digital and analog cameras shown in Figures 6.3 (a) and (b), respectively, were used for false alarm analysis. For mis-detection analysis, a number of targets of size 2×2 were added to these images. Low-stop and morphological filters were applied to these images. The outputs of these filters were fused using the two approaches described above. The local variance of the low-stop filter output, which is a measure of clutter, is shown in Figures 6.3 (c) and (d). The histogram of the local variance is shown in Figures 6.3 (e) and (f). It is seen that the analog camera image has a much higher clutter level than the digital camera image.

For the Constant False Alarm Rate (CFAR) fusion, the Locally Most Powerful (LMP) test was used. This gave the matched filter template as:

$$s = (g_l, g_m'(0))^t = (1.0, 0.8623)^t$$

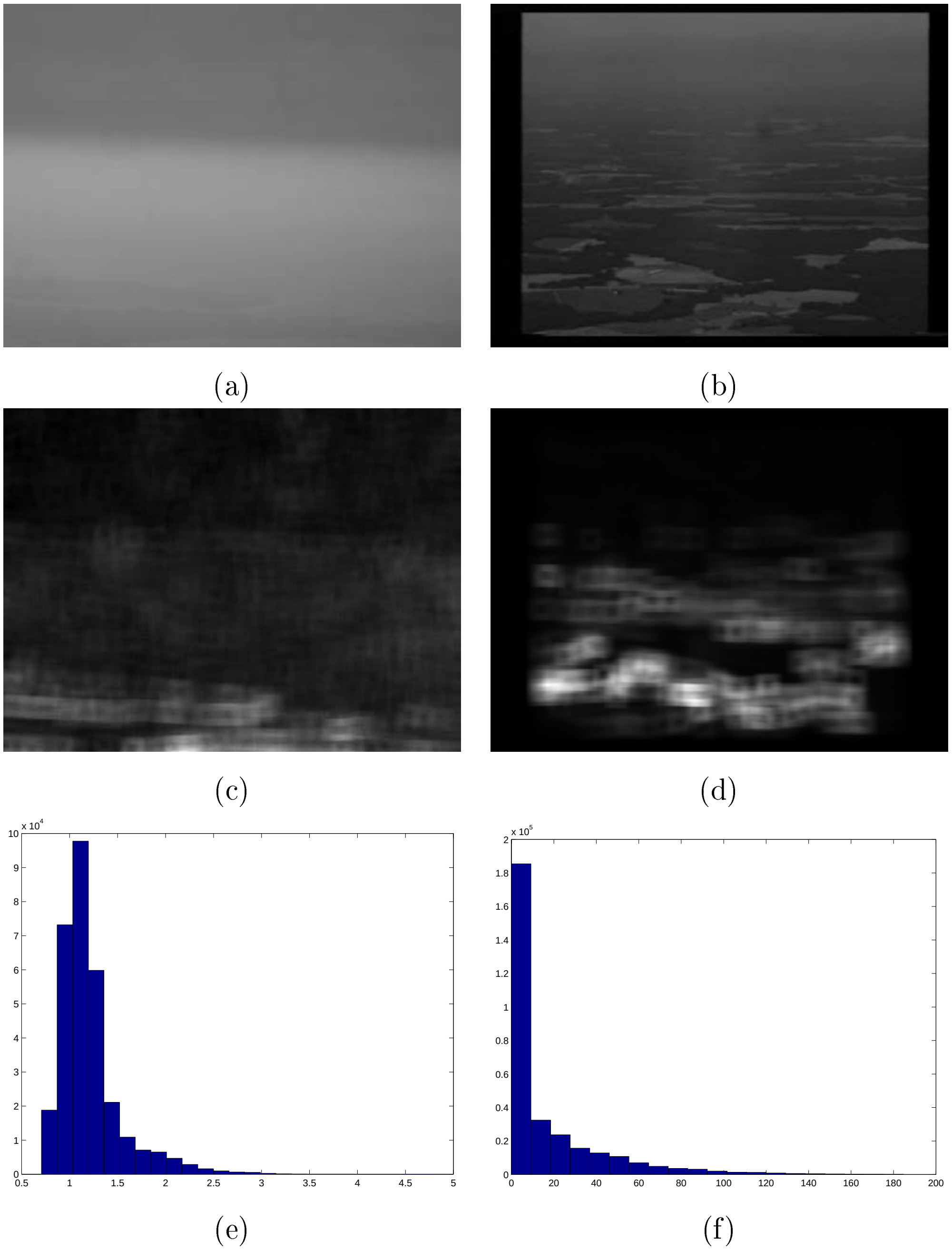
gives a slightly lower weight to the morphological filter when the level of noise is same for both the filter outputs. The plots of the mis-detections against the false alarms for the digital camera images are shown in Figure 6.4 (a) and (b), These use the assumption that the outputs of the low-stop and morphological filters are correlated. Algorithm fusion was also performed assuming independence between filters – i.e., $\rho = 0$. The independence assumption gave a slightly better performance for the fused filter as shown in Figure 6.4 (c) and (d), possibly because the correlation between filters may not have been adequately modeled. Similar plots using analog camera images are shown in Figure 6.5.

In both the cases, it is seen that the fused output does not give optimal performance for all the rates of false alarms. However, it can be observed that the fused output does give larger weight to the filter which has a better performance in the particular case. For example, in the case of digital camera images having relatively low clutter, (Figure 6.4), the better performing low-stop filter is given higher a weight. On the other hand, for analog camera images (Figure 6.5) with severe background clutter, the morphological filter which performs better is given a higher weight. Since the individual filter which would actually perform better in a particular case would not be known a-priori, the fusion at least serves the purpose of selecting the better filter.

To explore the reasons for the non-optimality of the CFAR approach, the method of thresholding the log likelihood ratio (LLR) was first used in place of the CFAR fusion. The results of thresholding likelihood ratio are shown in Figure 6.6. The outputs of the individual detectors, the likelihood ratio detector using each filter, and the fused likelihood ratio detector are shown for amplitudes of 6.0 and 8.0. The amplitude used for computing the likelihood ratio was of 6.0, which gave the signal template as:

$$\mu_\theta - \mu_0 = (6.0, 5.5603)^t = 6.0(1.0, 0.9267)^t$$

which is only slightly different from the LMP template (scaled), due to the non-linearity of the morphological filter.

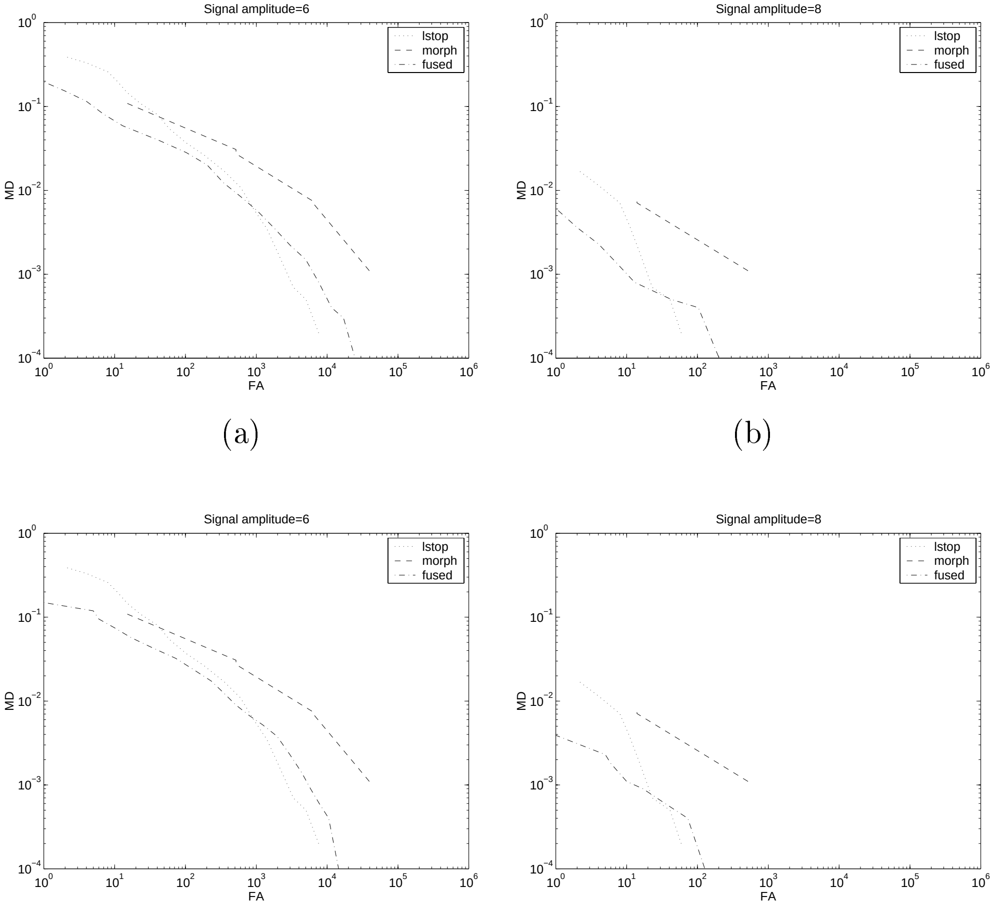


(a) (b)

(c) (d)

(e) (f)

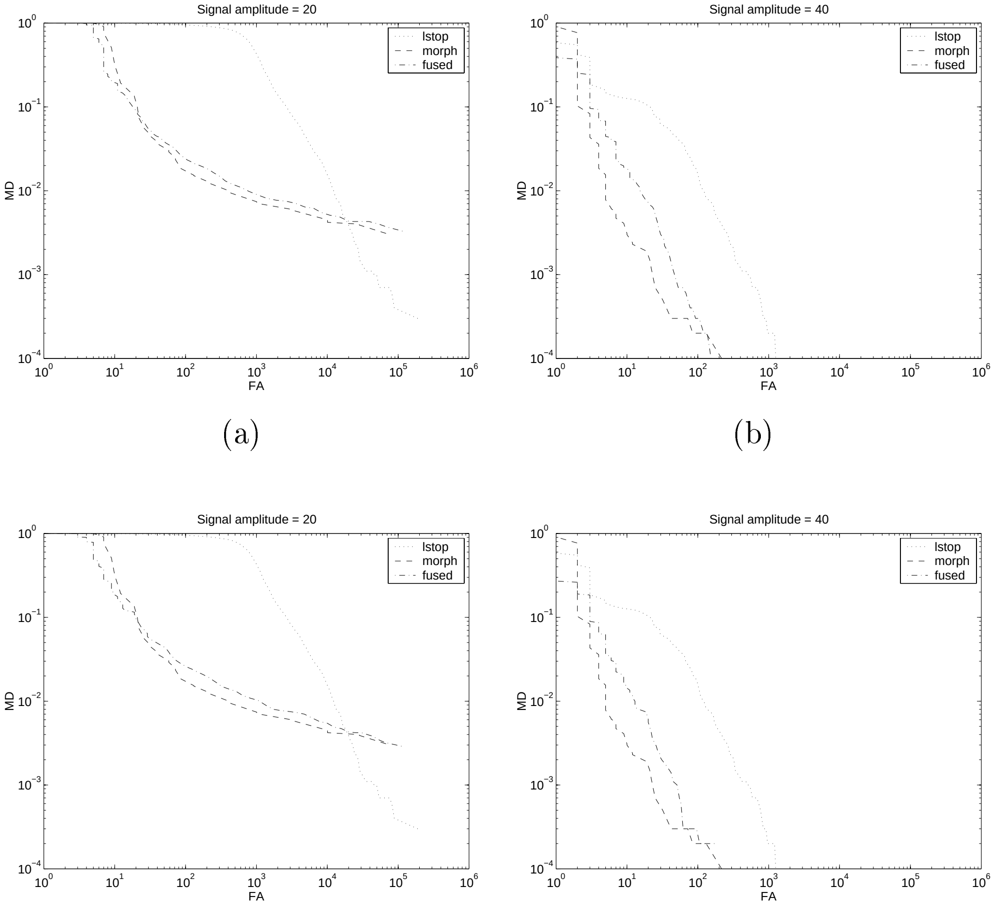
Figure 6.3: Images from (a) digital (b) analog camera with partly cluttered background. Image of the local variance of low-stop filter output, which is the measure of clutter for images from (c) digital (d) analog camera. Histogram of the local variance of low-stop filter output for images from (e) digital (f) analog camera.



(a) (b)

(c) (d)

Figure 6.4: Operating curves for digital camera image using CFAR fusion: Assuming correlation between filters with target amplitudes of: (a) 6.0 (b) 8.0 Assuming independence between filters with target amplitudes of: (a) 6.0 (b) 8.0



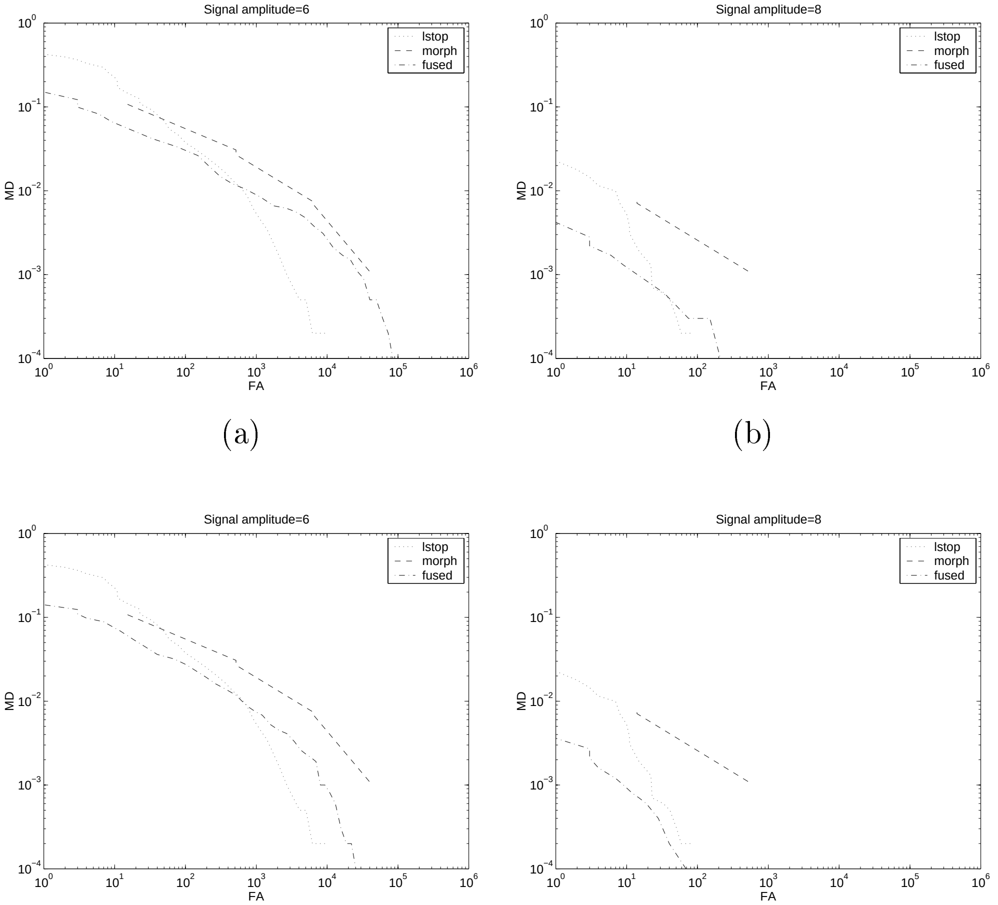
(a)

(b)

(c)

(d)

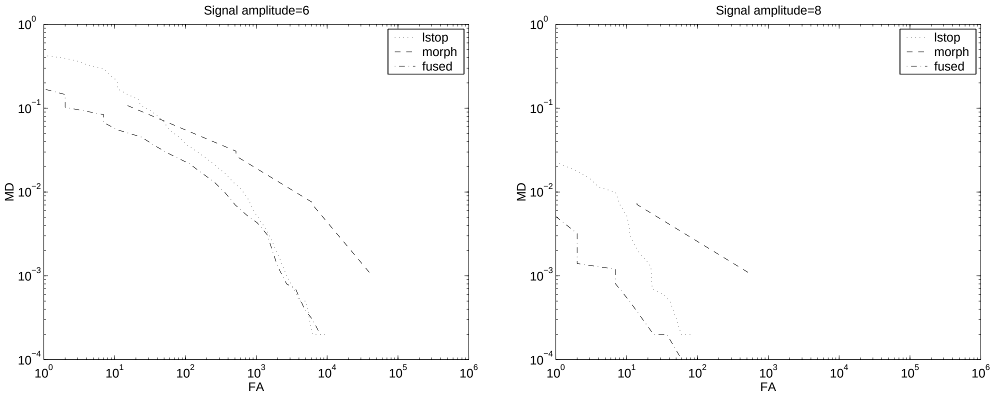
Figure 6.5: Operating curves for analog camera image using CFAR fusion: Assuming correlation between filters with target amplitudes of: (a) 20.0 (b) 40.0 Assuming independence between filters with target amplitudes of: (c) 20.0 (d) 40.0



(a) (b)

(c) (d)

Figure 6.6: Operating curves for digital camera image using LLR thresholding: Using correlation between filters with target amplitudes of: (a) 6.0 (b) 8.0 Assuming independence between filters with target amplitudes of: (c) 6.0 (d) 8.0



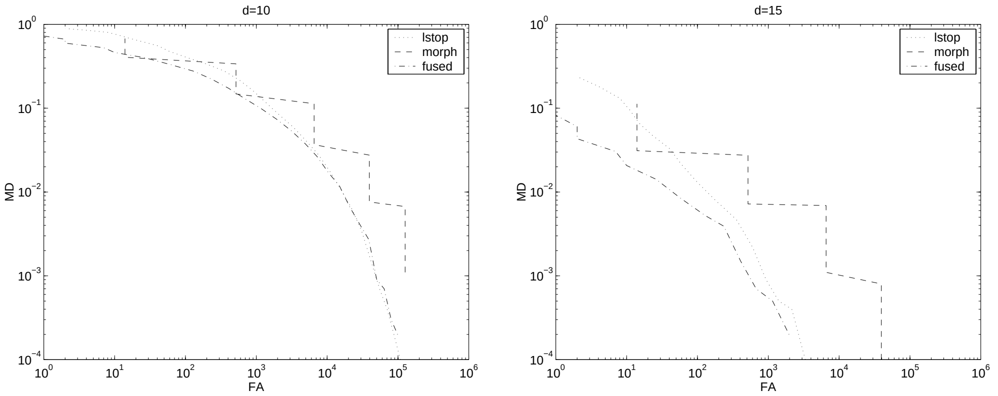
(a) (b)

Figure 6.7: Operating curves for digital camera image using LLR thresholding, and fixed value of morphological variance parameter $\sigma_m^2 = 1.5$, and assuming independence between the filters, for target amplitudes: (a) 6.0 (b) 8.0

However, it was seen that for the matched signal strength of 6.0, it still did not give desirable performance. Hence, another reason for this non-optimality was explored. It was observed that the variance parameter σ_m^2 of the morphological filter output was underestimated from the images. This unreliability of was because the estimation was performed using small windows around every point in the image. Furthermore, there was quantization error, since the noise in the images was of the same order as the gray level resolution of the real images. However, since the morphological filter is comparatively insensitive to clutter, the value of σ_m^2 remains approximately same throughout the image. Hence, the entire background image from the digital camera (without adding targets) was used to pre-compute the parameter value as $\sigma_m^2 = 1.5$. The low-stop filter parameter σ_l^2 was estimated as before, since its value *does* depend on the local clutter level. The correlation coefficient was assumed to be zero. The results obtained using these parameters are much better, and shown in Figure 6.7.

Hence, it can be concluded that the performance of CFAR approach was poor due to the following reasons:

1. CFAR fusion is not optimal under the condition of constant target amplitude.
2. The morphological filter parameters are not reliably estimated from small sized windows.



(a)

(b)

Figure 6.8: Operating curves for digital camera image using CFAR approach with condition of optimality, and fixed value of morphological variance parameter $\sigma_m^2 = 1.5$, assuming independence between the filters, where the targets have amplitude such that the deflection coefficient d is constant equal to: (a) 10.0 (b) 15.0

However, as shown before, the CFAR approach is theoretically optimal, when the target amplitude is not constant, but is adjusted so that the deflection coefficient d given by equation (6.17) remains constant. To check the optimality of the CFAR approach for this condition, another set of experiments was performed. The statistical parameters of the low-stop filter were estimated at every pixel using the background image without the addition of targets. The morphological filter parameters were estimated for the entire image (instead of individual pixels). Using the parameters of the low-stop and morphological filters, the deflection coefficient d_1 for a unit amplitude of the signal was computed at every pixel, and stored as a separate image. False alarm rate was also estimated using this image as before. For estimating mis-detection rates, targets were added to the background image. The amplitude of the target at a particular pixel was given by d/d_1 where d_1 is the function of the pixel coordinates and d is constant. The mis-detection rate was then estimated from a number of such images. The LMP template was used for fusing the outputs of the individual filters. The plots of the mis-detection rate against false alarm rate are shown in Figure 6.8. It can be seen that the fusion output is better or as good as the individual filter outputs, within experimental error.

Chapter 7

Detection of Translating Objects

In addition to the detection of objects on a collision course, it is useful to monitor the objects which are crossing the aircraft. For this purpose, a system was designed to specifically detect objects having a translational motion in the image. To distinguish translating objects from ground or cloud clutter, the following criteria were used:

1. The object should have sufficient signal strength.

2. The object should have an image velocity greater than a threshold.

3. The object should have a consistent motion – i.e., its velocity must not change abruptly.

The system to detect translating objects has been implemented on the pipelined image processing system, the DataCube MaxPCI described in Section 2.8 to obtain real time performance. The system was mounted on the Air Force Total In-Flight Simulator (TIFS/NC1314) aircraft, and flight tests were conducted by NASA with another aircraft flying in front of it. The detection and tracking of the target aircraft were demonstrated during the flight test.

This system is divided into two stages, an image processing stage and a tracking stage. The first stage consists of image processing steps which remove most of the clutter, and isolate potential features which could be translating objects. This stage involves repetitive image operations such as convolution, pointwise operations, histograms, etc. which are suitable for a pipelined architecture, and can be performed in integer format. Hence, these steps are implemented on the DataCube machine. The output of this stage is a list of image features which are likely to contain the target objects, including their positions and the signal strengths. However, the list may also contain features corresponding to background clutter, which are not separated by the simple image processing steps of the first stage. The second stage tracks these features to distinguish the genuine translating objects from background clutter using the criteria mentioned above. Since the first stage has reduced the volume of data to be operated on, more complicated target tracking algorithms can be implemented even on the host PC associated with the DataCube. The threshold used in the first stage is adjusted dynamically to give a nearly constant number of features for the second stage so that they can be processed in real time using the slower host. This matching of the output rate of one stage to the input rate of the next stage is known as the rate constraint criterion [5].

7.1 Image processing stage

This stage performs the basic image processing steps to suppress clutter and extract features which could potentially be translating targets.

1. Resolution Reduction: The resolution of the image is reduced so that the system is capable of operation in real time. The image is convolved with the following low-pass filter mask and then down-sampled by two in both horizontal and vertical directions.

$$M_0 = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 24 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 24 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

   Low-pass filtering suppresses high frequencies, which would otherwise have been aliased to low frequencies by the down-sampler. Although the image resolution is reduced, the signal to noise ratio is actually enhanced. This is because the target size is usually greater than 2 pixels, leading to spatial integration of the target contrast.

2. Low-stop filtering: A low-stop filter is applied to the reduced image to suppress background clutter. The filter is implemented by convolving the image with the following masks, one after the other:

$$M_1 = \frac{1}{64}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}, \; M_2 = -\frac{1}{128}\begin{bmatrix} 0 & 2 & 3 & 2 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 3 & 12 & -108 & 12 & 3 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 2 & 3 & 2 & 0 \end{bmatrix}$$

The mask M_1 is a smoothing mask, which performs spatial integration for large targets. A rectangular mask is used since the targets are expected to have a greater width than height. Application of the mask M_2 is equivalent to subtracting a smoothed image from the input image. The overall result of the two convolutions is the subtraction of a low-pass filter output with a larger mask from a low-pass filter output with a smaller mask. Hence, this step suppresses uniform background intensity and weak clutter corresponding to low frequencies, and also performs spatial integration for larger objects.

3. Image differencing: Image differencing is performed on the low-stop filtered images by subtracting consecutive frames. This is equivalent to a low-stop filter in temporal direction. Since the object is assumed to be translating, image differencing suppresses stationary objects corresponding to background clutter. It should be noted that steps 1 to 3 are theoretically interchangeable, since they are all linear filters. However, since these operations are performed with integer arithmetic of limited precision, the particular order of the steps is used to reduce the truncation error.

4. Non-maximal suppression: Directly using the output of the previous step would give rise to a large number of features for an extended target. Non-maximal suppression is performed to get a single feature (or sometimes a small number of features) for the entire target. Pixels can have both positive or negative values corresponding to bright and dark targets, respectively. Hence, an absolute value image is first formed, and every pixel which is not a local maximum in its 3×3 neighborhood is marked. The marked pixels are set to zero in the *original image* – i.e., the image before taking the absolute values.

5. Histogram formation: To extract candidate features, the output from the above steps should be thresholded. Furthermore, the threshold should be chosen so that the number of features neither overloads the tracking stage, nor keeps it unnecessarily idle. Hence, the threshold is selected so that the number of pixels exceeding the threshold is less than or equal to a fixed rate which matches the operation speed of the tracking stage. For this purpose, a histogram of the image is constructed. The threshold then is determined as the smallest pixel value for which the number of elements in the histogram bins above this value does not exceed the fixed rate. Applying this value as the threshold would then ensure that the number of features remains bounded.

6. Thresholding and feature output: Pixels in the image with the output value greater than the threshold are separated as features, and their positions as well as the amplitudes are transmitted to the tracking stage.

7.2 Tracking stage

This stage maintains a list of tracks containing the frame number, unique ID, position, velocity, and amplitude. The list is empty in the beginning. he following steps are repeated for every frame for which the list of features is received from the image processing stage:

1. Track update: For each track in the list of tracks, the list of features is scanned to obtain features in a neighborhood window around the track position. If one or more such features are found, the one with the largest amplitude is selected as the continuation of the track. Using the coordinates (z_1, z_2) of this feature, as well as the current track position (x_1, x_2) and velocity (u_1, u_2), the expected position and velocity for the next frame is estimated using a Kalman filter. The filter is applied separately for horizontal $(i = 1)$ and vertical $(i = 2)$ directions. For each direction, the state vector is given by $X_i = \left[\begin{array}{cc} x_i & u_i \end{array} \right]^t$, and the observation is the feature coordinate z_i. The track life n of the track is the number of frames in which the target has been observed, with adjustments made in the frames where the target is not observed. The measurement update is given by:

$$\begin{aligned} x_i^+(n) &= x_i(n) + K_1(n)\,(z_i - x_i) \\ u_i^+(n) &= u_i(n) + K_2(n)\,(z_i - x_i) \end{aligned} \tag{7.1}$$

The state update is given by:

$$\begin{aligned} x_i(n+1) &= x_i^+(n) + u_i^+(n) \\ u_i(n+1) &= u_i^+(n) \end{aligned} \tag{7.2}$$

The Kalman filter matrix $K(n) = \left[\begin{array}{cc} K_1(n) & K_2(n) \end{array} \right]^t$ is pre-computed using the inverse covariance formulation of the Kalman filter. The computation is performed for a number of $n = 1 \ldots N$, where N is large enough so that $K(N)$ does not change significantly with N.

The track amplitude is updated using recursive averaging according to the following equation:

$$F(n+1) = f(n) + \alpha F(n) \tag{7.3}$$

where $F(n)$ and $F(n+1)$ are the track amplitudes for the current and next frames, $f(n)$ is the feature amplitude, and α is the forgetting factor. The track life n is incremented by one.

If no feature satisfying the above conditions is found in the neighborhood of the track, the position and velocity are extrapolated using only the state update. Theoretically, this would mean that the values of the Kalman filter matrix would have to be recomputed. To avoid such a computation, the value of the track life n is reduced by a factor to approximately simulate the effect of having 'lost track' of the feature. The feature amplitude is updated using $f(n) = 0$ in equation (7.3).

2. Formation of new tracks: After all the current tracks are updated, features in the feature list are used to check for new tracks. For every feature, the list of tracks is scanned to see if a track is already there in its neighborhood. If not, a track is created out of the feature with its track life $n = 1$. Its position (x_1, x_2) will be the same as feature position (z_1, z_2), whereas velocity (u_1, u_2) is initialized to zero. The actual velocity will be computed only in the next frame.

3. Pruning the list of tracks: If the number of tracks is too large, the stage can get overloaded and fail to operate in real time. To eliminate this possibility, if the number of tracks are greater than a particular number, the weakest tracks are deleted.

4. Merging similar tracks: It may happen that two or more tracks may be formed corresponding to the same object. Hence, tracks which are very close to each other and have nearly the same velocity are merged, retaining the one with the larger track amplitude.

5. Output: Tracks which satisfy the criteria of the object, including having an amplitude larger than a threshold, as well as other factors are output as potential objects.

7.3 Results

The real-time image capturing, recording, and processing system were demonstrated by the flight tests conducted by NASA. During the first set of flight tests, image sequences were captured and recorded successfully at the rate of 30 frames per second. The tracking algorithms were designed and fine-tuned using these image sequences. During the next set of flight tests, in addition to the real-time capturing and recording, the translating target tracking algorithm was executed concurrently at the rate of 15 frames per second. Several

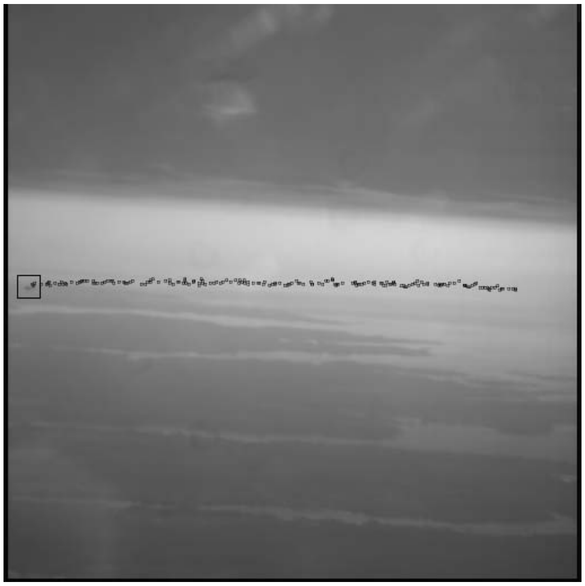


Figure 7.1: Tracking algorithm applied on an image sequence with the target aircraft translating from right to left at a distance of 3 nautical miles. The target aircraft is located at the end of the track in this image.

image sequences with the target aircraft crossing the host aircraft were obtained. It was observed that the system successfully detected and tracked the translating object during the flight tests. Figure 7.1 shows a trace of the tracking algorithm applied on an image sequence with the target aircraft translating from right to left at a distance of 3 nautical miles.

Table 7.1 summarizes the performance of the translating target tracking algorithm with different distances between the host and the target aircraft, during the first set of flight tests. The false alarm rate is measured as the ratio of the total number of false alarms throughout the sequence to the number of image frames in the sequence. The mis-detection rate is measured as the ratio of the number of frames in which the target was missed to the total number of frames. The false alarm rate depends on the amount and motion of clutter in the images, whereas the mis-detection rate depends on the target size and contrast, and therefore increases with the target distance in most cases. Since false alarms can be very annoying to the pilots, a low false alarm rate was more desirable than a low mis-detection rate. Hence, the parameters of the algorithms were selected to reduce the false alarm rate, and were same for all the scenarios. It is possible to get a better performance by adjusting parameters according to the characteristics (such as the clutter level) of each scenario.

Table 7.1: The performance of the translating target detection algorithm for a number of target distances. The false alarm rate is the ratio of the total number of false alarms throughout the sequence to the number of image frames in the sequence. The mis-detection rate is the ratio of the number of frames in which the target was missed to the total number of frames.

Distance (nmi)	Mis-detection rate	False alarm rate
1.5	0.061	0.000
1.8	0.113	0.000
2.0	0.394	0.000
2.4	0.059	0.000
3.0	0.056	0.000
4.7	0.335	0.183
5.0	0.803	0.147
5.4	0.643	0.000

The performance was relatively poor in the cases where the host aircraft rotated about its own axes, resulting in large image motion of background features. To improve the performance, the image motion due to aircraft rotation should be compensated using the aircraft navigation data. If this data is unavailable, the background motion should be modeled to separate independent object motion. For example, Irani and Anandan [9] separated the scene motion into planar and parallax components, and identified independently moving objects having a significant parallax. However, since the DataCube architecture is capable only of simple image processing operations, any such procedure would have to be performed on the host machine, using a feature based approach.

Chapter 8

Conclusion

This research was focused on designing and implementing algorithms for detection of obstacles in the flight path of the aircraft using the image sequences obtained from the on board cameras. The main contributions of this research and the possible avenues of future work are described below.

8.1 Contributions of this research

- Basic algorithms performing signal enhancement were tested for detecting flying objects using the image sequences provided by NASA. Performance characterization of these algorithms was conducted using simulated and real image sequences. It was observed that the algorithms performed well on images with little or no clutter, but their performance degraded in presence of clutter.

- To distinguish the objects on a collision course from the background clutter, the difference in the behavior of their image translation and expansion were studied. Conditions under which these criteria are useful were derived. Novel methods for estimating the rates of image translation and expansion over long image sequences were designed and tested on the image sequence with a large amount of background clutter. The approach successfully separated the obstacle from the clutter.

- Algorithm fusion to overcome limitations of algorithms was studied, and it was observed that under proper conditions, a combination of algorithms performed better than the individual algorithms.

- A real-time system using pipelined image-processing hardware was designed to detect objects crossing the aircraft. The tracking algorithm to separate background clutter from crossing objects was developed and implemented on the host machine associated with the system.

8.2 Future work

- Many of the research ideas, such as the use of translation and expansion, algorithm fusion, etc. were tested individually. The future goal would be to combine these into an integrated system for obstacle detection. Performance characterization of this system could be done with more real image sequences.

- During the estimation of image translation to discriminate a hazardous object from background clutter, the compensation of aircraft rotation was performed using the navigation system information. Use of background clutter to model the aircraft motion could be explored, so that the compensation could be performed even without the navigation system information.

- False expansion occurring due to the rotation of the target aircraft can be studied. This expansion takes place only in a particular direction, resulting in deformation of the object in the image. Methods to distinguish this deformation from uniform expansion can be studied.

- Gaussian models were used for studying the behavior of individual algorithms to perform algorithm fusion. Better models could be developed, especially in presence of clutter, where the Gaussian models would not be as robust.

- To improve the performance of crossing object detection, the image motion due to aircraft rotation should be compensated. This could be done either using the navigation data from the aircraft, or by modeling the image motion separate independent object motion. Since the DataCube architecture is capable only of simple image processing operations, such a procedure should be performed on the host machine, using a feature based approach.

Bibliography

[1] N. Ancona and T. Poggio. Optical flow from 1-D correlation: Application to a simple time-to-crash detector. *International Journal of Computer Vision*, 14:131–146, 1995.

[2] J. Arnold, S. Shaw, and H. Pasternack. Efficient target tracking using dynamic programming. *IEEE Trans. on Aerospace and Electronic Systems*, 29(1):44–56, January 1993.

[3] Y. Baram and Y. Barniv. Obstacle detection by recognizing binary expansion patterns. *IEEE Trans. on Aerospace and Electronic Systems*, 32(1):191–197, January 1996.

[4] Y. Barniv. Dynamic programming solution for detecting dim moving targets. *IEEE Trans. on Aerospace and Electronic Systems*, 21(1):144–156, January 1985.

[5] J. S. Bird and M. M. Goulding. Rate-constrained target detection. *IEEE Trans. on Aerospace and Electronic Systems*, 28(2):491–503, April 1992.

[6] D. Casasent and A. Ye. Detection filters and algorithm fusion for ATR. *IEEE Trans. on Image Processing*, 6(1):114–125, January 1997.

[7] E. Francois and P. Bouthemy. Derivation of qualitative information in motion analysis. *Image and Vision Computing*, 8(4):279–288, November 1990.

[8] G. C. Holst. *CCD Arrays, Cameras and Displays.* JCD Publishing, Winter Park, FL, 1996.

[9] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(6):577–589, June 1998.

[10] T. Kanungo, M. Y. Jaisimha, J. Palmer, and R. M. Haralick. A methodology for quantitative performance evaluation of detection algorithms. *IEEE Trans. on Image Processing*, 4(12):1667–1674, December 1995.

[11] R. Kasturi, O. Camps, L. Coraor, K. Hartman, T. Gandhi, and M.-T. Yang. Performance characterization of target detection algorithms for aircraft navigation. Technical report, Dept. of Computer Science and Engineering, The Pennsylvania State University, October 1998.

[12] R. Kasturi, Y.-L. Tang, and S. Devadiga. A model-based approach for detection of runways and other objects in image sequences acquired using an on-board camera. Technical Report CSE-94-051, Department of Computer Science and Engineering, Penn State University, August 1994.

[13] S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory.* Prentice Hall, Upper Saddle River, NJ, 1993.

[14] S. S. Krause. *Avoiding Mid-Air Collisions.* TAB books, Mc Graw Hill Inc., Blue Ridge Summit, PA, 1995.

[15] R. C. Nelson and J. Y. Aloimonos. Obstacle avoidance using flow field divergence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(10):1102–1106, 1989.

[16] K. Nishiguchi, M. Kobayashi, and A. Ichikawa. Small target detection from image sequences using recursive max filter. In *Proc. SPIE*, volume 2561, pages 153–166, July 1995.

[17] H. V. Poor. *An Introduction to Signal Detection and Estimation.* Springer-Verlag, New York, NY, 2nd edition, 1994.

[18] S. M. Tonissen and R. J. Evans. Performance of dynamic programming techniques for track-before-detect. *IEEE Trans. on Aerospace and Electronic Systems*, 32(4):1440–1451, October 1996.

[19] D. Wood. *Jane's World Aircraft Recognition Handbook.* Jane's Information Group Ltd., Coulsdon, UK, 1992.